

UNIVERSITE JOSEPH FOURIER – GRENOBLE I
U.F.R EN INFORMATIQUE
ET MATHEMATIQUES APPLIQUEES



Master de Recherche 2ème année
Intelligence, Interaction et Information

Projet présenté par :
Gilles Bailly

Sous la responsabilité de :
Laurence Nigay CLIPS-IMAG, Grenoble 1
David Auber LABRI, Bordeaux 1

TITRE : **Interaction bi-manuelle pour la manipulation de graphes biologiques**

Présenté et soutenu le 22 juin 2005

Jury : Mme Joëlle Coutaz
M. Jerôme Euzenat
M. Yves Denneulin
Examinateur extérieur :
Mme Sabine Coquillart



Laboratoire de Commnunication Langagière
Et Interaction Personne-Système
Fédération IMAG
Unité Mixte de Recherche : CNRS – UJF - INPG



Remerciements

Comment ne pas commencer par Laurence qui m'a fait confiance et qui m'a permis de découvrir un domaine passionnant de l'informatique. Je la remercie vivement de m'avoir proposé ce sujet, de m'avoir encadré pendant toute cette année et de m'avoir poussé à réaliser tant de choses captivantes.

Merci à David de m'avoir si bien accueilli à Bordeaux et de toute l'aide qu'il m'a apportée.

Merci également à tous les membres de l'équipe IIHM pour l'ambiance de travail et leur aide, en particulier : Joëlle, Lionel, Gaétan, François et vincent.

Merci à Romain Boussard pour ses dessins.

Je remercie mes parents, Marie-Pierre et mes amis pour leur soutien permanent.

Table des matières

CHAPITRE I INTRODUCTION **7**
1 MOTIVATIONS 7
2 OBJECTIFS ET DEMARCHE DE TRAVAIL 8
3 CONTEXTE 8
3. 1 Projet NavGraphe *8*
3. 2 Position par rapport au projet *9*
4 STRUCTURE DU MEMOIRE 9

Partie I Espace de conception pour l'interaction bi-manuelle

CHAPITRE II INTERACTIONS BI-MANUELLES **15**
1 CANAL GESTUEL 15
1. 1 Gestes *15*
1. 2 Canal gestuel *16*
1. 3 Conclusion sur le canal gestuel *16*
2 PSYCHOLOGIE DE L'ACTION BI-MANUELLE : Y. GUIARD 17
2. 1 Analogie : la chaîne cinématique *17*
2. 2 Coopération entre les deux mains à la lumière de l'analogie avec la chaîne cinématique *18*
2. 2. 1 Contexte défini par la main non dominante 18
2. 2. 2 Complémentarité des deux mains 19
2. 2. 3 Précédence de la main non dominante 19
2. 2. 4 Dominance de la main dominante 19
2. 3 Conclusion sur la psychologie *20*
3 AVANTAGES ET INCONVENIENTS DE L'INTERACTION BI-MANUELLE 21
3. 1 Avantages *21*
3. 1. 1 Avantages manuels 21
3. 1. 2 Avantages cognitifs 21
3. 2 Inconvénients *22*
3. 3 Conclusion sur les avantages et les inconvénients de l'interaction bi-manuelle *22*
4 CONCLUSION SUR L'INTERACTION BI-MANUELLE 23

CHAPITRE III ESPACES DE CONCEPTION EXISTANTS **24**
1 DISPOSITIFS 24
1. 1 Taxonomie des dispositifs physiques selon Buxton *24*
1. 2 Espace de conception des dispositifs physiques de Mackinlay, Card et Robertson *26*
1. 3 Taxonomie des tâches graphiques selon Foley *28*
1. 4 Conclusion sur les dispositifs *29*
2 MODALITE D'INTERACTION ET MULTIMODALITES 31
2. 1 Définition : modalité d'interaction et multimodalité *31*
2. 2 Espace de FROLICH *32*
2. 3 Espace MSM *33*
2. 4 Propriétés CARE *37*
2. 5 Propriétés CARE étendues [Vernier 01] *37*
2. 6 Conclusion sur la multimodalité *38*
3 ESPACE DE CONCEPTION DES INTERACTIONS BI-MANUELLES 40
3. 1 Action discrète, Action continue *40*
3. 2 Temporalité des actions *41*
3. 2. 1 Temporalité dans l'interaction à une main 42
3. 2. 2 Temporalité dans l'interaction deux mains 42
3. 3 Conclusion sur l'interaction bi-manuelle *43*

4 CONCLUSION SUR LES ESPACES DE CONCEPTION 44

CHAPITRE IV NOUVEL ESPACE DE CONCEPTION BI-MANUEL 2M 45

1 MULTIMODALITE ET INTERACTION BI-MANUELLE : NIVEAUX D'ABSTRACTION 45
2 INTERACTION BI-MANUELLE ET MULTIMODALITE : ASPECTS TEMPORELS 46
3 NOUVEL ESPACE DE CONCEPTION DE L'INTERACTION BI-MANUELLE : 2M 49
3. 1 Apports d'un espace de conception 49
3. 2 Dimensions de l'espace 2M 50
3. 3 Dimension Dépendance 51
3. 3. 1 Dépendance d'un point de vue utilisateur 51
3. 3. 2 Dépendance d'un point de vue système 53
3. 3. 3 Conclusion sur la dimension dépendance 54
3. 4 Apports de l'espace 2M 54
3. 5 Pouvoir taxinomique..................... 54
3. 5. 1 Défilement et sélection 55
3. 5. 2 Editeur de dessin [Yee 04]..................... 56
3. 5. 3 Toolglass 57
3. 5. 4 Utilisation de touches spéciales 58
3. 5. 5 Interfaces palpables 59
3. 6 Pouvoir génératif et comparatif..................... 60
4 CONCLUSION SUR L'ESPACE 2M..................... 61

Partie II Application : manipulation bi-manuelle de graphes biologiques

CHAPITRE V APPLICATION DE NOS RESULTATS A LA MANIPULATION BI-MANUELLE DE GRAPHES BIOLOGIQUES..................... 65

1 STRUCTURES BIOLOGIQUES MANIPULEES 65
2 DEMARCHE DE COLLABORATION AVEC LES BIOLOGISTES..................... 66
2. 1 Analyse descendante..................... 66
2. 2 Analyse ascendante..................... 67
3 TACHES IDENTIFIEES 67
3. 1 Tâche d'édition..................... 67
3. 2 Tâche de comparaison..................... 68
3. 3 Tâche d'évolution 68
3. 4 Tâche de visualisation multiple 68
4 CONCLUSION SUR LA COLLABORATION AVEC LES BIOLOGISTES 68

CHAPITRE VI TECHNIQUES D'INTERACTION CONÇUES ET DEVELOPPEES 69

1 DESCRIPTION DE L'APPLICATION..................... 69
1. 1 Arbre des tâches 69
1. 2 Techniques d'interaction conçues 73
1. 2. 1 BiToolKit : boîte à outils bi-manuelle 73
1. 2. 2 Fonctionnalité de navGraphe..................... 76
2 CONCEPTION LOGICIELLE 79
2. 1 Modèle d'architecture PAC-Amodeus 79
2. 2 Architecture du système NavGraphe 81
2. 3 Solution architecturale générique pour l'interaction bi-manuelle 83
2. 3. 1 Fusion lexicale..................... 84
2. 3. 2 Fusion syntaxique..................... 84
2. 3. 3 Fusion sémantique..................... 84
3 DEVELOPPEMENT 84
3. 1. 1 Architecture globale : PAC-Amodeus 85
3. 1. 2 Patrons de conception..................... 86
3. 1. 3 Optimisations..................... 87
3. 1. 4 Une classe, deux fichiers 87
3. 1. 5 Documentations..................... 88
3. 1. 6 Installation..................... 88
3. 1. 7 Conclusion sur le développement..................... 88
4 CONCLUSION SUR LES TECHNIQUES CONÇUES ET DEVELOPPEES..................... 88

CHAPITRE VII CONCLUSION..................... 90

1 SYNTHESE DE NOTRE CONTRIBUTION 90
2 PERSPECTIVES 92
2. 1 Extensions........ 92
2. 2 Prolongements........ 92
3 BILAN PERSONNEL DE L'EXPERIENCE DE RECHERCHE DE CETTE ANNEE........ 93
CHAPITRE VIII ANNEXES 99
1 CVISION/TCLVISION 99
1. 1 TCLVision........ 99
1. 2 CVision 100
2 INSTALLATION ET DEVELOPPEMENT........ 100
2. 1 Installer Linux 101
2. 2 Installer la carte graphique nvidia........ 101
2. 3 Installer Tulip 102
2. 4 Installer Eclipse........ 102
2. 5 Installer CDT (module C et C++ pour eclipse) 103
2. 6 Installer glut, glu et opengl........ 103
2. 7 Installer SlimLib 103
2. 8 Installer le projet navGraphe 103
3 STORY-BOARD 105
4 PROJET RICM2 107

Chapitre I Introduction

Le projet de recherche a trait à la conception ergonomique et à la réalisation logicielle de techniques d'interaction bi-manuelle avec de grands graphes. L'étude en prenant en compte la spécificité des données manipulées, vise à contribuer à l'amélioration de l'utilisabilité des systèmes qui exploitent des graphes. De très nombreux domaines d'application reposent sur la manipulation, l'exploration et la navigation dans de grands graphes comme la recherche dans les bases de données vidéo, l'exploration de cartes ou réseaux sémantiques et la visualisation de structures biologiques. Notre domaine d'application est la biologie et nous traitons des structures biologiques, en particulier l'ARN (acide ribonucléique).

1 Motivations

Plus que jamais, il faut assister l'utilisateur pour explorer, naviguer et retrouver l'information dont il a besoin dans des graphes. Le problème classique en conception d'interface homme-machine (IHM) est d'identifier, pour un utilisateur donné accomplissant une tâche avec un système, une représentation des concepts et une façon d'interagir avec ces derniers, qui soient en accord avec l'attente et les compétences de l'utilisateur. Dans ce contexte, notre étude focalise sur l'interaction bi-manuelle pour des tâches de manipulation de graphes.

Nous sommes partis de l'observation que l'homme utilise naturellement au quotidien les deux mains pour effectuer des tâches dans le monde physique. Or dans le contexte de l'informatique, les interfaces sont très généralement à une main et lorsqu'elles sont bi-manuelles, elles sont souvent mal conçues et adaptées [Scott 01]. Par exemple considérons les interfaces standards souris et touches spéciales du clavier. Nous avons environ trois boutons (gauche, centre, droit) sur la souris et quinze touches spéciales sur le clavier (SHIFT, CTRL, ALT, TAB, ←, ↑, →, ↓, etc.). Donc la main dominante (généralement la main droite) qui utilise la souris a seulement trois boutons, alors que la main non dominante (généralement la main gauche) en possède quinze. Or la main dominante devrait traiter une plus grande quantité d'informations que la main non dominante.

Le sujet de recherche se focalise donc sur l'interaction bi-manuelle. Les avantages de l'interaction bi-manuelle semblent intuitivement évidents tant pour les novices pour le côté naturel, que pour les experts en augmentant le pouvoir d'expression et l'efficacité. Néanmoins, comme pour la multimodalité, l'interaction bi-manuelle peut s'avérer moins utilisable que l'interaction à une seule main. Par exemple dans [Kabbash 94], une tâche de dessin (choix de la couleur et coloriage de lignes) a permis de conclure que les résultats peuvent être moins bons par rapport à une technique à une main. Cela peut s'expliquer par un coût cognitif important.

Il parait donc intéressant de caractériser finement l'interaction bi-manuelle afin de pouvoir proposer des techniques d'interaction en accord avec l'utilisateur.

2 Objectifs et démarche de travail

Comme annoncé, notre étude concerne l'interaction bi-manuelle : **l'originalité de notre approche consiste à considérer l'interaction bi-manuelle comme multimodale en entrée** : une modalité d'entrée par main. Pour cela, nous identifions les étapes de travail suivantes :

- Etudier l'état de l'art sur l'interaction bi-manuelle et la multimodalité afin d'abord d'évaluer dans quelle mesure nous pouvons rapprocher ces deux domaines et ensuite d'informer et affiner chacun des domaines à l'aide de l'autre.
- Organiser les aspects de conception de l'interaction bi-manuelle au sein d'un canevas cohérent et unificateur : Cet espace doit être utile pour classer les systèmes bi-manuels, pour concevoir de nouvelles techniques d'interaction et pour évaluer ces techniques.
- Appliquer ces résultats au domaine de la biologie et en particulier à la manipulation de graphes d'ARN. Les biologistes et les informaticiens en IHM sont deux communautés qui se connaissent peu. Cela implique deux démarches complémentaires :
    - Ascendante : proposer de nouvelles techniques d'interaction sur de grands graphes pour les biologistes.
    - Descendante : Observer et analyser les besoins des biologistes pour leur offrir des techniques d'interaction en accord avec leurs besoins et leur contexte d'usage.

3 Contexte

3. 1 Projet NavGraphe

Le sujet de recherche s'inscrit dans le contexte du projet NavGraphe (ACI Masses de Données).

Le projet NavGraphe vise à combiner les méthodes automatiques de l'algorithme de graphes, les requêtes de bases de données et l'œil expert de l'utilisateur pour une navigation et une exploration performante de graphes. Le projet NavGraphe se focalise sur trois domaines d'application où les graphes sont manipulés :

- Bases de données vidéo : Les données étant des séquences d'images, les capacités du cortex humain se révèlent bien meilleur que la recherche textuelle. L'enjeu pour ce domaine est de pouvoir réaliser une indexation des séquences vidéo en temps réel.

- Cartes sémantiques : Les cartes sémantiques sont un moyen d'établir une indexation sur un système d'information. Les techniques algorithmiques pour manipuler, classer, regrouper, rechercher des informations dans les cartes sémantiques définissent les défis de ce domaine.

- Structures biologiques : Les techniques les plus avancées pour la conception de nouvelles protéines, pour l'étude de structures tels que l'ADN, l'ARNm ou les réseaux cellulaires passent par une étape de visualisation sous forme de graphes. Les bio-chimistes ou généticiens sont demandeurs d'outils de visualisation et d'interaction pour manipuler et naviguer dans ces très grosses structures

3. 2 Position par rapport au projet

Notre domaine de compétence dans NavGraphe est l'IHM (Interaction-Homme Machine) pour la navigation, l'exploration et la manipulation de grandes quantités d'informations. Nous nous appuyons sur un logiciel de visualisation de graphes du LaBRI : Tulip [Auber 01].

Dans le cadre de notre projet, les données manipulées sont seulement des structures biologiques et en particulier de l'ARNm. Cependant, les techniques d'interaction proposées dans ce mémoire peuvent être adaptées facilement à d'autres types de données.

L'étude s'inscrivant dans un projet à plus long terme qui sera continué au-delà de l'année Master 2 Recherche, l'objectif était de fournir des résultats réutilisables tant en conception (résultats applicables à d'autres domaines d'application que la manipulation d'ARN) qu'en développement (architecture globale et règle de programmation pour de grands projets).

4 Structure du mémoire

Notre mémoire s'organise en deux parties, l'une dédiée à **la conception de l'interaction bi-manuelle** et l'autre à **l'application de nos contributions au cas de la manipulation de graphes biologiques**. La première partie est composée de trois chapitres.

Le premier chapitre définit l'interaction bi-manuelle d'un point de vue psychologique. Il comprend quatre paragraphes :

Le paragraphe 1 reprend la notion de canal gestuel afin de situer l'interaction bi-manuelle dans un cadre d'étude plus vaste.

Le paragraphe 2 fait le point sur les résultats de l'action bi-manuelle en psychologie. Nous listons un ensemble de propriétés qui qualifie la coopération des deux mains pour l'accomplissement d'une tâche.

Le paragraphe 3 présente les avantages et les inconvénients de l'interaction bi-manuelle selon les points de vue actionnel et cognitif.

Le paragraphe 4 fait une synthèse de l'action bi-manuelle afin d'aborder les différents espaces de conception dans le chapitre suivant.

Le second chapitre présente les espaces de conception en informatique qui nous ont amené à élaborer notre propre espace de conception. Les trois domaines investigués font l'objet d'un paragraphe.

Le paragraphe 1 expose les différents espaces de conception pour les dispositifs car il existe un lien important entre le dispositif utilisé et la main qui le manipule.

Le paragraphe 2 traite de la multimodalité, l'originalité de notre approche étant de considérer l'interaction bi-manuelle comme multimodale en entrée.

Le paragraphe 3 dresse un état de l'art des espaces de conception pour l'interaction bi-manuelle.

Le paragraphe 4 fait une synthèse des différents espaces de conception existant en informatique et nous permet d'aborder notre nouvel espace de conception dans le chapitre suivant.

Le troisième chapitre de ce mémoire définit l'espace solution. Il présente nos contributions et expose notre nouvel espace de conception pour l'interaction bi-manuelle : 2M. Il se décompose en quatre paragraphes.

Le paragraphe 1 enrichit les travaux sur l'interaction multimodale pour prendre en compte le rôle des deux mains et introduit ainsi des niveaux d'abstraction pour l'étude de l'interaction bi-manuelle.

Le paragraphe 2 étudie l'interaction à deux mains au regard de la multimodalité.

Le paragraphe 3 est consacré à présenter notre espace de conception : 2M qui permet de prendre en compte les aspects temporels mais aussi de dépendance de la multimodalité lors de la conception de techniques d'interaction bi-manuelle. Une fois les axes de l'espace exposés, nous analysons le pouvoir taxinomique, génératif et comparatif de l'espace 2M.

Le paragraphe 4 propose une synthèse des contributions et de l'espace 2M.

La deuxième partie du mémoire vise à valider nos résultats en les mettant en œuvre dans la conception et la réalisation logicielle d'un système interactif bi-manuel. Le domaine d'application est la manipulation bi-manuelle de graphes d'ARN par des biologistes. Elle est composée de trois chapitres :

Le quatrième chapitre rappelle le problème biologique sur lequel nous voulons appliquer nos résultats sur l'interaction bi-manuelle. Il est décomposé en 4 paragraphes.

Le paragraphe 1 présente les structures biologiques manipulées.

Le paragraphe 2 expose notre démarche de collaboration avec les biologistes afin d'identifier leurs besoins mais aussi leur présenter des outils possibles en adéquation avec leur travail.

Le paragraphe 3 identifie les résultats de la collaboration avec les biologistes à travers quatre tâches.

Le paragraphe 4 fait une synthèse sur les besoins identifiés et nous permet grâce aux tâches identifiées d'aborder dans le cinquième chapitre la réalisation logicielle.

Le cinquième chapitre de ce mémoire est dédié à la réalisation logicielle : les différentes techniques d'interaction conçues et développées.

Le paragraphe 1 présente les techniques d'interaction conçues

Le paragraphe 2 traite de la conception logicielle. Il expose l'architecture logicielle de l'application, correspondant à la phase de conception logicielle. De plus, nous détaillons comment l'interaction bi-manuelle est traitée dans l'architecture logicielle.

Le paragraphe 3 est dédié à la phase d'implémentation et présente les différentes règles suivies pour fournir un code réutilisable.

Le paragraphe 4 conclue ce chapitre par un bilan de la conception.

En conclusion, nous soulignons nos contributions théoriques et réalisation du système NavGraphe et développons des perspectives à des travaux.

Partie I

Espace de conception pour l'interaction bi-manuelle

Chapitre II Interactions bi-manuelles

Dans ce chapitre, nous étudions rapidement le canal gestuel afin de situer l'interaction bi-manuelle dans un cadre d'étude plus général et de spécifier quel est le type de geste étudié dans ce mémoire. Nous exposons ensuite les résultats issus de la psychologie pour mieux cerner les capacités bi-manuelles humaines et les différentes propriétés psychologiques à prendre en compte. Cette étude nous permet ensuite d'étudier les avantages et les inconvénients de l'interaction bi-manuelle. Enfin, nous conclurons sur l'interaction bi-manuelle d'un point de vue humain. Ceci nous permettra d'aborder dans les chapitres suivants les espaces de conception en informatique et en particulier les espaces de conception pour l'interaction bi-manuelle.

1 Canal gestuel

Cette partie est dédiée au geste et au canal gestuel afin de bien cerner notre étude et les gestes étudiés dans le cadre d'étude général du geste humain.

1. 1 Gestes

Dans [Kurtenbach 90], *un geste est un mouvement du corps qui contient l'information*. Faire un mouvement de main pour dire au revoir est un geste. Appuyer sur une touche sur un clavier n'est pas un geste parce que le mouvement du doigt pour aller frapper une touche n'est ni observé ni significatif. Tout ce qui importe est quelle touche fut pressée. De même, un concepteur esquissant avec un crayon sur le papier n'est pas considéré comme étant en train de faire un geste parce que ce n'est pas le mouvement et le maintien de la main qui sont observés. Tout ce qui importe est la ligne qui est tracée. Cependant, il convient de noter que dans certains systèmes interactifs, la façon de réaliser ce geste est captée (comme la pression du stylet qui est traduite par une épaisseur de trait).

[Hummels 98] utilise alors la définition suivante : *un geste est un mouvement de son corps qui donne à lui-même ou à un associé dans la communication.* Cet associé peut être un humain ou un partenaire.

[Cadoz 94] distingue trois types de geste suivant leur finalité principale:

- Le geste sémiotique : il sert à faire connaître par le moyen de signes (ex : les gestes accompagnant la parole).
- Le geste épistémique : c'est un moyen de connaître (ex : la combinaison de la perception cutanée avec le mouvement des mains et des doigts qui permet de percevoir la forme ou la taille d'un objet).
- Le geste ergotique : il permet d'agir physiquement sur les objets pour les saisir, les déplacer, les modeler.

Dans le cadre de ce projet, les gestes envisagés sont de nature ergotique : ils servent à manipuler des graphes. La façon de réaliser ce geste (pression, rapidité) ne sera pas dans un premier temps capté par le système informatique.

1. 2 Canal gestuel

Le canal gestuel comprend les gestes de la main dominante (la main droite pour les droitiers) et la main non dominante (la main gauche pour les droitiers), les gestes des doigts, des lèvres (handicapés, ou tâches bien spécifique telle que le pilotage d'avion), des pieds (pédales), des membres ou le corps tout entier pour commander la machine. Dans le cadre de l'Interaction Homme-Machine, le canal gestuel est encore sous-exploité.

1. 3 Conclusion sur le canal gestuel

Dans cette étude, nous nous intéressons dans le cadre du canal gestuel aux gestes ergotiques des deux mains (dominante et non dominante). La façon dont les gestes sont réalisés n'est pas captée, seule la reconnaissance des gestes effectués pour spécifier une commande ou une tâche élémentaire sera étudiée.

2 Psychologie de l'action bi-manuelle : Y. Guiard

L'homme utilise naturellement sa main gauche et sa main droite de concert en leur confiant des rôles nettement différenciés. La psychologie s'est intéressée à une relation de compétition entre les deux mains et s'est attardée à quantifier la supériorité de la main dominante en oubliant de prêter attention à une possible logique de division inter-manuelle du travail.

Y. Guiard adopte un point de vue original : son modèle de la **chaîne cinématique** focalise sur la coopération entre les deux mains pour accomplir un but.

The two hands cooperate with each other in the service of action.
[Guiard 87]

La chaîne cinématique décrit ce travail coopératif entre les deux mains : la main non dominante agit la première pour définir le cadre de travail dans lequel la main dominante effectue l'action souhaitée (par exemple, pour un droitier, la main gauche tient la feuille de papier sur laquelle la main droite écrit).

Dans les paragraphes suivants, nous exposons les résultats les plus importants pour l'interaction homme-machine de la chaîne cinématique (Kinematic Chain KC).

2. 1 Analogie : la chaîne cinématique

La chaîne cinématique est définie comme l'interaction au sein d'un système de composants : ces composants concourent à résoudre un problème posé au niveau du système. Le terme « chaîne cinématique », qui provient d'un sous-domaine de la mécanique, dénote l'assemblage de plusieurs liens rigides mis en série. Par exemple, le bras dans la globalité peut être vu comme un ensemble de liens : le bras, l'avant bras, la main, et le doigt et un ensemble d'articulations : l'épaule, le coude, le poignet et les articulations des doigts. Ces éléments travaillent ensemble pour résoudre un but commun.

Les quatre propriétés intéressantes de KC sont :

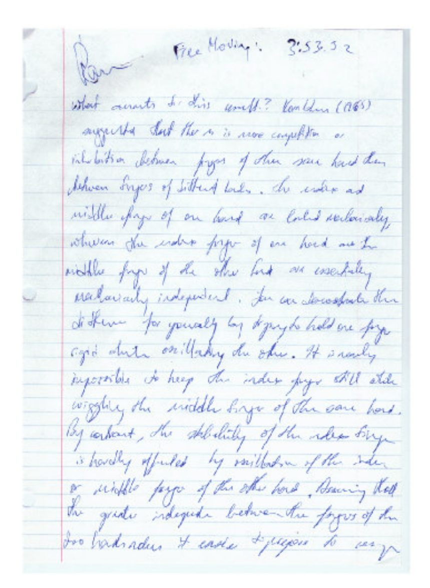
- Armature de référence. Les éléments distales (terminal), les mains, travaillent dans le contexte fournit par les éléments proximales (origine), l'épaule. Par exemple, la rotation du poignet se fait en fonction de l'avant bras et non pas de la main.
- Hiérarchisation des composants. Par exemple, lorsque l'on étend le bras pour saisir un objet, le bras travaille beaucoup plus que la main. Le bras travaille de manière macro métrique tandis que la main fait des mouvements plus petits et plus précis de niveau micrométrique.
- Organisation temporelle des activités des différents composants. En effet, en considérant le même exemple, le bras fait le mouvement de s'approcher de l'objet, bien avant celui de la main ou du doigt. C'est ce dernier qui terminera le mouvement.
- Dominance de l'élément distal. Dans l'exemple du bras, l'élément terminal c'est-à-dire la main a plus d'importance. Dans le geste de lancer une balle, on dit bien lancer une balle à la main alors que tout le bras participe à l'action, mais c'est elle qui touche la balle.

2. 2 Coopération entre les deux mains à la lumière de l'analogie avec la chaîne cinématique

Le système coopératif à deux mains peut être comparé aux composants proximaux et distaux d'une chaîne cinématique. Les quatre grands problèmes énoncés dans KC : l'armature de référence, la différentiation métrique, la priorité et la dominance peuvent être appliquées au comportement humain à deux mains.

2. 2. 1 Contexte défini par la main non dominante

Dans la plupart des manipulations, la main dominante travaille en référence à la main non dominante. Par exemple, la main dominante dévisse le bouchon de la bouteille que la main non dominante tient, de mêmela main dominante essaye d'enfiler le fil dans l'aiguille que la main non dominante tient. Ceci est toujours valide pour marteler ou visser une visse. Pour toutes ces opérations, les deux mains sont nécessaires et travaillent ensemble de manière dissymétrique. Manifestement, la main non dominante fixe le contexte, tandis que la manipulation proprement dite est effectuée par la main dominante. La main non dominante a donc un rôle important pour préserver la position et l'orientation d'un objet malgré les perturbations induites par la main dominante.



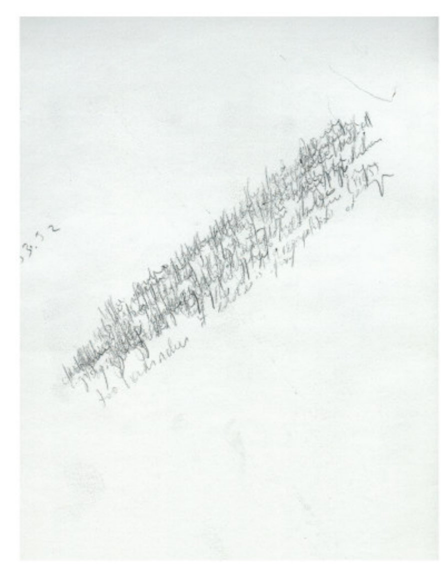


Figure 1 : Deux vues du même texte écrit à la main [Guiard 87]. La page du côté gauche de la Figure 1 montre ce que le scribe a écrit sur le papier. La page du côté droit Figure 1montre ce qui est apparu sur la copie au carbone placée sur la table en dessous de la feuille écrite avec la main dominante. L'angle de l'écriture, comme le recouvrement des lignes est une conséquence du changement de contexte effectué par la main non dominante.

Ce résultat de la main non dominante qui fixe le contexte d'évolution de la main dominante est illustré avec l'exemple de l'écriture de la Figure 1. Contrairement à la croyance commune, l'écriture normale implique les deux mains, Il a été observé expérimentalement [ATH 84]que l'écriture est ralentie lorsque l'on utilise qu'une seule main. Les mains gauches et droites fonctionnent à deux niveaux d'action :

- Le premier correspondant au mouvement du stylo dans le contexte de référence de la page,
- Le second au mouvement de la feuille de papier.

Guiard a montré [Guiard 87] que les auteurs habiles replacent périodiquement leur feuille de papier avec la main non dominante, afin de limiter l'extension (les mouvements macro métrique) de la main dominante. Il convient de noter, toujours par analogie de KC qu'un contexte de référence n'a pas besoin d'être stationnaire, celui-ci peut être dynamique et défini par la main non dominante.

2. 2. 2 Complémentarité des deux mains

À la lumière du modèle de KC, les mains gauche et droite sont considérées comme deux organes complémentaires, un macro métrique et un micro métrique, qui coopèrent ensemble. En première analyse, le concept d'un contraste micro-macro n'ajoute rien à l'observation commune que les mouvements de la main dominante sont généralement supérieurs c'est-à-dire à la fois plus forts et plus précis que ceux de la main non dominante. Dans le cas de l'écriture, la main dominante est associée à une résolution plus fine au niveau spatial et temporelle (petite oscillation du stylo), tandis que la main non dominante consiste à une résolution plus large (grands mouvements de la feuille, plus espacés dans le temps). Dans [Kabbash 93], il a été montré d'une expérience de sélection de cible que pour de petites distances et petites cibles, la main dominante était supérieure. Cependant, pour de plus grandes cibles et de plus grandes distances, les deux mains était à peu près identique. L'expérience prouve que la main non dominante est plus qu'une approximation faible de la main dominante. Les mains sont complémentaires, chacune a ses propres forces et faiblesses.

2. 2. 3 Précédence de la main non dominante

Lorsque l'on utilise un microscope, on actionne d'abord le bouton macro métrique, puis on affine après avec le bouton micrométrique. De même pour atteindre une cible éloignée, on commence d'abord par actionner les articulations macro métriques comme l'épaule et le coude, puis lorsque l'on se trouve à proximité de la cible, les articulations des poignets et des doigts accomplissent les micros ajustements finaux. Le modèle KC suggère que ce principe s'applique également dans le contexte du mouvement à deux mains. Ainsi, toutes les fois où la main non dominante remplit une fonction de stabilisation, elle doit intervenir avant que la main dominante agisse. Par exemple, pour enfoncer un clou, celui-ci doit être tenu au bon endroit et avec la bonne orientation, lorsque la collision avec le marteau se produit. De même pour la rédaction d'un texte, la main droite commencera à écrire seulement après que la main gauche ait correctement placé la feuille.

2. 2. 4 Dominance de la main dominante

Les deux facettes de la dominance de la main sont la préférence et la supériorité de la main dominante. La supériorité a très largement été étudiée et observée par des tests expérimentaux en laboratoires. Le cas de la préférence est plus compliqué. Alors que les psychologues la résumaient par l'asymétrie des mains, Guiard apporte un regard nouveau. Dans le modèle KC, la

main représente l'élément terminal du bras, c'est elle qui finit les actions, qui touchent le monde. L'interaction à deux mains peut être modélisée comme une KC où la main dominante représente le terminal de la chaîne. Cette main reçoit plus de considération car elle est l'élément le plus remarquable dans ce système à deux mains. Le composant directement en contact de l'environnement termine des actions typiquement initiées par son associé.

2. 3 Conclusion sur la psychologie

Pour notre étude, nous retenons du modèle de Y. Guiard que l'interaction à deux mains suit le modèle de la chaîne cinématique et donc les quatre propriétés suivantes :

- La main non dominante fixe le contexte, tandis que la manipulation proprement dite est effectuée par la main dominante.
- La main non dominante est macro métrique tandis que la main dominante est micrométrique.
- La main non dominante intervient avant que la main dominante agisse.
- La main dominante est la main préférée car elle termine les actions et touche le monde.

Nous allons maintenant noter les avantages et les inconvénients de l'interaction à deux mains dans le cadre de l'informatique.

3 Avantages et inconvénients de l'interaction bi-manuelle

Dans la partie précédente, nous avons étudié comment interagissaient les deux mains grâce à un modèle psychologique. Néanmoins ce modèle ne nous indique pas quand il faut utiliser l'interaction bi-manuelle. En première analyse, il semble évident que l'interaction bi-manuelle est avantageuse tant pour son caractère naturel, que pour son pouvoir d'expression. Nous allons donc affiner ces avantages, classés en deux catégories : manuels, cognitifs. Puis nous verrons que des expérimentations ont aussi révélé des inconvénients.

3. 1 Avantages

Dans [Leganchuck 98], deux types d'avantages de l'interaction bi-manuelle sont identifiés : manuels et cognitifs.

3. 1. 1 Avantages manuels

Les avantages manuels viennent de l'efficacité accrue du rapport mouvements-temps. Une des raisons est que l'utilisateur peut utiliser deux fois plus de degrés de liberté simultanément. L'autre raison est que l'interaction bi-manuelle permet à des utilisateurs d'accomplir des tâches plus rapidement. La hausse du rapport mouvements-temps peut être attribuée à plusieurs choses. D'abord, quelques opérations peuvent être accomplies en parallèle (tel que se déplacer et mesurer un rectangle simultanément). Ensuite l'économie de temps peut résulter d'une réduction de la quantité de commutation de tâche (baisse du temps de transition entre les tâches). Par exemple, deux points finaux d'une ligne peuvent être déplacés sans devoir déplacer le curseur dans les deux sens entre eux. Nous pouvons alors se référer à la loi de Fitts qui met en relation le temps et la distance de la cible.

3. 1. 2 Avantages cognitifs

Les avantages cognitifs sont liés à la réduction de la charge mentale de l'utilisateur. Il existe deux types d'avantages cognitifs.

Les dispositifs bi-manuels correspondent à une métaphore du monde réel (caractère naturel). En effet, au quotidien, nous utilisons nos deux mains pour réaliser une tâche comme l'expliquait [Guiard 87]. L'intention est d'utiliser cette métaphore pour assister l'apprentissage d'un style d'interaction. Par exemple, les techniques d'interaction reposant sur les ToolGlasses et Magic Lenses [Bier 94] sont fondés sur la métaphore du peintre qui tient sa palette dans une main tandis qu'il dessine ou pointe avec l'autre main. Un autre avantage de l'interaction bi-manuelle est de diminuer la charge de composition et de visualisation de la tâche courante à un niveau anormalement bas imposé par des techniques uni-manuelles. En effet, lors de la réalisation d'une tâche (comme déplacer, mesurer et tourner un objet graphique) l'utilisation de techniques uni-manuelles implique pour l'utilisateur l'élaboration d'un plan d'actions plus complexes qu'avec des techniques bi-manuelles. L'utilisateur n'a plus besoin de composer, de penser et de projeter les étapes élémentaires d'une tâche. De plus, le retour d'information rapide de la

manipulation implique un traitement de la tâche à un niveau plus élevé : L'utilisateur voit immédiatement le résultat de l'action par rapport au but qu'il veut atteindre.

Les actions épistémiques sont effectuées pour découvrir l'information qui est cachée ou difficile à calculer mentalement. Les actions physiques rendent souvent le calcul cognitif interne plus facile, plus rapide et plus fiable. Par exemple, nous utilisons parfois nos doigts quand nous comptons, les joueurs d'échecs débutant peuvent physiquement déplacer temporairement une pièce d'échecs à sa nouvelle position pour évaluer le mouvement et les retours offensifs possibles par un adversaire. Les actions épistémiques peuvent améliorer la connaissance en réduisant (1) la mémoire, (2) nombre d'étapes et (3) probabilité des erreurs impliquées pendant le calcul mental. Il semble que l'interaction à deux mains offre plus d'occasions épistémiques comparées aux techniques à une main. C'est parce que l'interaction bi-manuelle fournit en soi plus de degrés de liberté. Il est donc possible de manoeuvrer des données plus rapidement pour découvrir l'information qui est cachée ou difficile à calculer mentalement. Ceci a finalement comme conséquence des temps plus rapides d'accomplissement de tâche [Kirsh 94].

Nous venons d'exposer des avantages de l'utilisation de l'interaction bi-manuelle qui sont d'ordre manuel et cognitif. Nous en étudions maintenant les inconvénients.

3. 2 Inconvénients

Il a été constaté que l'exécution à deux mains n'est pas toujours meilleure dans la pratique. Pour s'en convaincre, considérons l'exemple classique suivant : "se taper la tête avec une main et tourner l'autre main sur l'estomac". L'interaction à deux mains peut donc être un fardeau pour l'utilisateur par la difficulté à coordonner les deux mouvements. Par exemple dans [Kabbash 94], plusieurs techniques (uni-manuelles et bi-manuelles) sont étudiées expérimentalement pour une tâche de dessin : choix de la couleur et coloriage de lignes. L'évaluation permet de conclure que lorsque chaque main est assignée à des tâches indépendantes, les résultats sont moins bons par rapport à une technique à une main. Ce résultat est expliqué par un effort cognitif accru et une division de l'attention plus importante. De plus, dans le cas où un utilisateur réalise une tâche compliquée, l'interaction bi-manuelle peut se révéler moins utile car la tâche force l'utilisateur à coordonner les actions de ses deux mains en plus de la grande difficulté de la réalisation de la tâche.

3. 3 Conclusion sur les avantages et les inconvénients de l'interaction bi-manuelle

En synthèse, nous retenons deux types d'avantages de l'interaction bi-manuelle : avantages manuels et cognitifs. Pour les avantages manuels, nous distinguons une efficacité accrue du rapport mouvements-temps qui peut être expliquée par une hausse du degré de liberté, une baisse du temps de transition entre les tâches et un certain parallélisme entre les tâches. Pour les avantages cognitifs, nous distinguons une réduction et une externalisation des chargements de planification et de visualisation, une exécution de la tâche à un niveau de granularité supérieure.

Néanmoins, en phase de conception la prudence s'impose et des expérimentations ont souligné la charge cognitive et actionnelle de synchronisation des deux mains.

4 Conclusion sur l'interaction bi-manuelle

Dans ce chapitre, nous avons étudié les résultats les plus importants sur les caractéristiques du geste humain. Nous retenons pour notre étude, les points suivants :

- Le type de geste effectué par les mains est ergotique.
- La main non dominante fixe le contexte, tandis que la manipulation proprement dite est effectuée par la main dominante.
- La main non dominante est macro métrique tandis que la main dominante est micrométrique.
- La main non dominante intervient avant que la main dominante agisse.
- La main dominante est la main préférée car elle termine les actions et touche le monde.
- L'interaction bi-manuelle n'est pas dans l'absolu meilleure que l'interaction uni-manuelle.

Chapitre III Espaces de conception existants

Tandis que le chapitre précédent est consacré aux résultats issus de la psychologie, nous étudions dans ce chapitre les espaces de conception issus du domaine de l'interaction Homme-Machine.

Nous étudions d'abord les espaces de conception dédiés aux dispositifs d'interaction puisque l'utilisateur peut manipuler des dispositifs de caractéristiques différentes dans chaque main. Considérant l'interaction bi-manuelle comme multimodale, nous jetons un nouveau regard sur l'interaction bi-manuelle en exposant ensuite les espaces de référence pour la conception multimodale. Nous finissons ce chapitre par l'exposé du seul espace de conception pour l'interaction bi-manuelle. Nous concluons par une synthèse des espaces de conception étudiés.

1 Dispositifs

Les dispositifs d'entrées sont les outils qui permettent à l'homme d'interagir avec le système. Un dispositif se caractérise par plusieurs critères comme la rapidité de déplacement, la précision, une bonne ergonomie pour une prise en main rapide, mais aussi un apprentissage rapide. Cette liste n'est pas exhaustive. Pour classer les dispositifs actuels, il existe trois taxonomies consacrées aux périphériques d'entrée. Dans l'ordre chronologique :

- L'approche de Buxton en 1983.
- L'approche de Foley, Wallace et Chan en 1984.
- L'approche de Mackinlay, Card et Robertson en 1990.

Nous allons présenter ces trois taxonomies (manière de caractériser un dispositif), en commençant par la taxonomie de Buxton qui est la plus orientée dispositif. Puis nous verrons l'approche de Mackinlay qui constate des manques dans la taxonomie de Buxton et qui les comble en proposant une démarche mathématique. La troisième et dernière approche est celle de Foley qui est de plus haut niveau d'abstraction et orienté sur les tâches graphiques. Ces espaces de conception n'ont pas été établis au départ pour l'interaction à deux mains. Notre but sera de montrer dans quelle mesure les dispositifs à deux mains (un dispositif dans chaque main) peuvent être caractérisés grâce à ces espaces de conception.

1. 1 Taxonomie des dispositifs physiques selon Buxton

Selon Buxton, un *dispositif d'entrée* est un transducteur de propriétés physiques que l'on peut caractériser dans un espace à trois dimensions. Comme le montre la Figure 2,

- le premier axe de l'espace de classification (*Property sensed*) dénote les propriétés que le dispositif est capable de capter. Ces propriétés sont au nombre de trois : la position, la pression, le mouvement ;
- le second axe correspond au nombre de dimensions captées pour chaque propriété (*Number of Dimensions*). Par exemple, pour la souris, la position est déterminée

par deux valeurs prises dans un système de coordonnées. Pour le manche à balai, la position est un triplet ;

- le troisième axe distingue le type direct ou indirect de capture (*Sensing type*). Par exemple, pour un écran tactile ou une caméra, la mesure est directe (T) alors que la souris implique un intermédiaire mécanique (M).

		Number of Dimensions				
		1	2	3		
Property sensed	Position	Rotary Pot, Sliding Pot	Tablet, Light Pen, Joystick	3D Joystick	M	Sensing type
			Touch Tablet, Touch Screen		T	
	Motion	Continuous Rotary Pod, Treadmill, Thumbwheel	Trackball	Trackball	M	
		Tasa Ferinstat	Tasa X-Y Pad		T	
	Pressure	Torque Sensing, Pressure Pad	Isometric Joystick		T	

Figure 2 : Dispositif physique vu comme transducteur de trois propriétés physiques [Buxton 83]

Les deux premiers axes de cette taxonomie définissent les degrés de liberté du transducteur dans l'espace. Ils expriment la capacité du dispositif à traduire les actions de l'utilisateur en une forme exploitable par le système informatique. La richesse du dispositif est alors fonction du nombre de propriétés physiques captées et pour chaque propriété, le nombre de dimensions spatiales. Le type (direct ou indirect) de la capture, traduit le degré d'engagement physique de l'utilisateur. A la Figure 3, nous positionnons plusieurs exemples de dispositifs au sein de la taxonomie de la Figure 2.

Ici, ne sont considérés que les dispositifs actionnables par le geste instrumental : le microphone et la caméra ne sont pas évoqués. Ces derniers constituent cependant des cas intéressants : une caméra peut servir à reconnaître un geste [Berard 92]. Par exemple, pour l'interaction bi-manuelle, la caméra permet de suivre des jetons que manipule l'utilisateur avec ses deux mains.



Figure 3 : Classification de quelques dispositifs dans la taxonomie de Buxton

1. 2 Espace de conception des dispositifs physiques de Mackinlay, Card et Robertson

Mackinlay, Card et Robertson partent du constat que l'assemblage de dispositifs élémentaires en unités de contrôle plus complexes est une dimension importante en conception. L'objectif est alors de définir une théorie qui systématise la connaissance et le raisonnement sur les dispositifs d'entrée.

Le domaine de cette théorie est l'ensemble des dispositifs d'entrée. Chaque *dispositif* est défini par un sextuplet <M, In, Out, R, S, W>. Les opérateurs de la théorie expriment la possibilité d'assembler plusieurs dispositifs :

- **M** est un opérateur de manipulation ("Manipulation operator") appliqué par l'utilisateur. Il désigne une propriété physique captable par le dispositif (par exemple, une position).
- **In** est le domaine des valeurs d'entrée possibles de M. Par exemple, l'ensemble des réels compris entre 0 et 180 degrés pour un bouton rotatif.
- **Out** est le domaine des valeurs de sortie possibles du dispositif. Si le bouton comporte 5 positions (cas d'un bouton cranté), Out contient alors les 5 valeurs angulaires correspondantes.
- **R**, ou fonction de résolution ("Resolution function"), définit la correspondance entre les domaines In et Out. Dans le cas du bouton, cette fonction exprime les conditions de passage entre les angles du mouvement effectué par l'utilisateur et les crans (angles en sortie) imposés par le dispositif.
- **S** dénote l'état actuel du dispositif ("state"). Il comprend la valeur *In* de l'entrée prise dans In à l'instant présent t, la valeur *Out* prise dans Out et un état interne.
- **W** désigne le fonctionnement externe et interne du dispositif physique ("Works"). Il permet de modéliser des relations intéressantes entre In et Out en fonction de l'état. Par exemple, certains dispositifs captent des valeurs sans percevoir les valeurs intermédiaires entre deux valeurs captées successives. La tablette est dans ce cas : il y a des gestes humains que le dispositif ne peut percevoir d'où un risque de perte d'informations potentiellement utiles à la qualité de la communication.

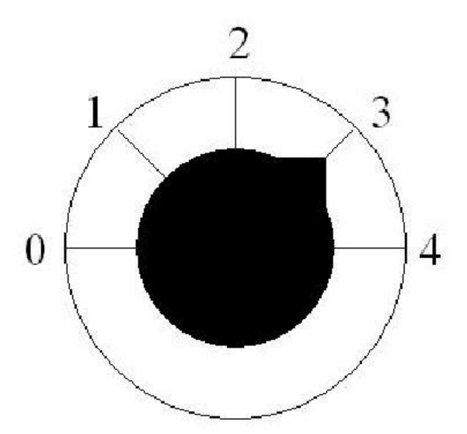


Figure 4 : Bouton cranté

Par exemple, le bouton cranté de la Figure 4 est décrit par le sextuplet suivant :

M = Rz, rotation par rapport à l'axe z

In = [0°, 180°]

Out = [0°, 45°, 90°, 135°, 180°]

R = *f* avec f(In) =

[0°, 22.5°] → {0°}

[22.5°, 67.5°] → {45°}

[67.5°, 112.5°] → {90°}

[112.5°, 157.5°] → {135°}

[157.5°, 180°] → {180°}

S = θ

W = {}

Le sextuplet <M, In, Out, R, S, W> dénote un dispositif primitif. Mackinlay et ses co-auteurs poposent des opérateurs de compositions pour modéliser des dispositifs plus complexes. Pour notre étude, un opérateur nous semble particulièrement intéressant : la fusion. Formellement, la fusion est le produit cartésien de deux dispositifs sur les domaines d'entrée des dispositifs. Par exemple, la position d'une souris est le produit cartésien des deux dispositifs élémentaires captant respectivement l'ordonnée et l'abscisse. En considérant les dispositifs manipulés par chaque main nous pouvons grâce à l'opérateur fusion caractériser un dispositif composé et bi-manuel. Ce point sera étendu dans le chapitre suivant consacré à notre espace de conception.

En conclusion, Mackinlay et ses coauteurs systématisent la description des dispositifs d'entrée au moyen d'une algèbre simplifiée. L'avantage est double : élargir le champ de conception à la prospection de nouveaux dispositifs et ouvrir la voie à l'évaluation du pouvoir d'expression des dispositifs en relation avec la tâche. Cette taxonomie comme celle de Buxton, prend le dispositif comme élément directeur. Toutes deux permettent, en considérant les caractéristiques d'un dispositif donné de s'interroger sur son adéquation à une tâche donnée. La taxonomie de Foley présentée au paragraphe suivant, prend le point de vue inverse : étant donné une tâche, elle vise à proposer un ensemble de dispositifs idoines.

1. 3 Taxonomie des tâches graphiques selon Foley

La taxonomie de Foley adopte la tâche comme point d'ancrage. Pour classer un dispositif, Foley propose de le mettre en relation avec les tâches graphiques qu'il permet d'accomplir. Ces tâches sont au nombre de cinq : sélection, position, orientation, chemin, quantificateur et texte qui peuvent, selon les dispositifs être accomplies de manière directe ou indirecte. La Figure 5 présente la taxonomie de Foley.

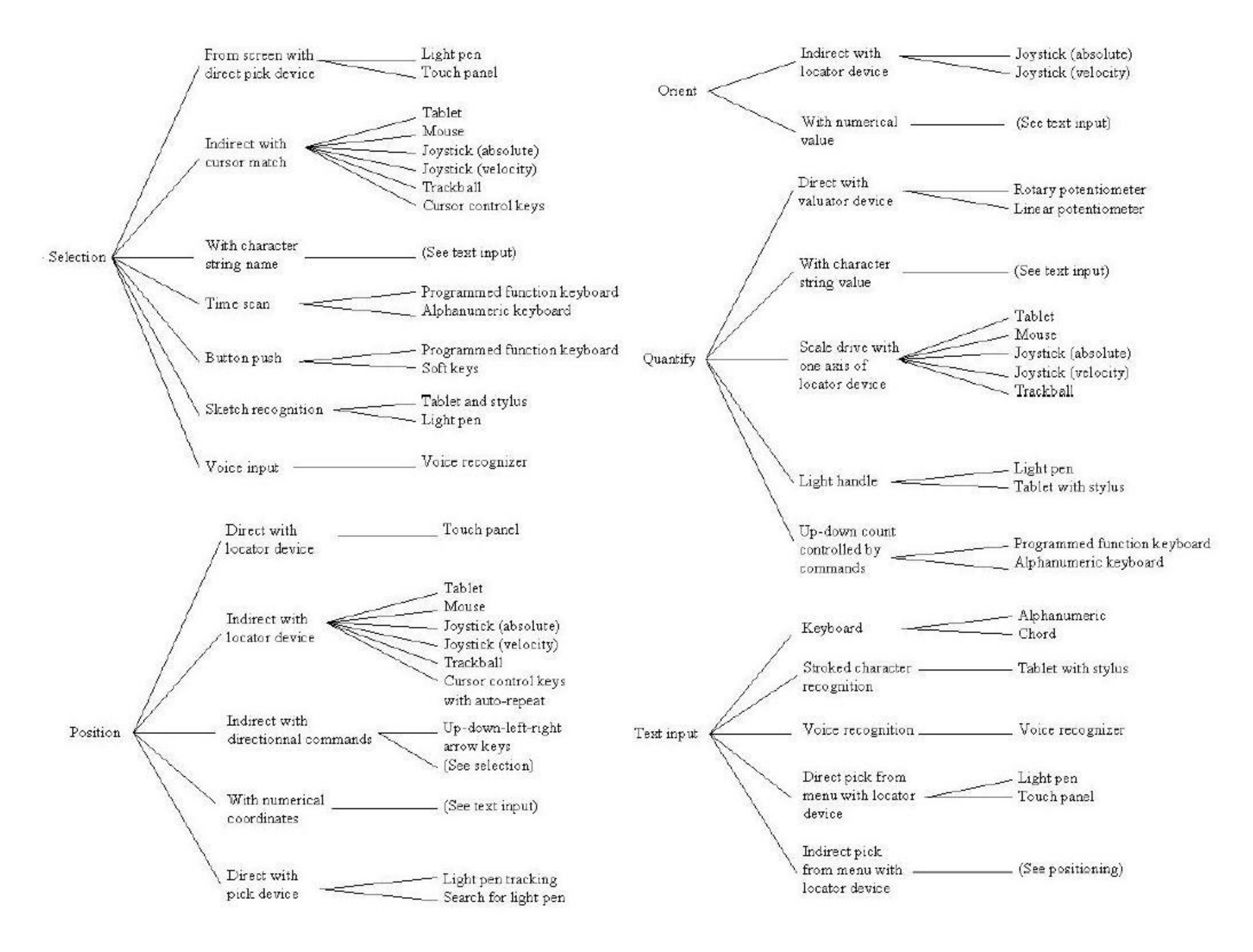


Figure 5 : La taxonomie de Foley dirigée par les tâches graphiques [Foley 84]

Contrairement aux taxonomies précédentes qui organisent les dispositifs en fonction des propriétés physiques qu'ils sont capables de modéliser, Foley dirige la classification par la tâche. On peut y voir comme avantage, des recommandations prescriptives sur le choix des dispositifs en fonction des tâches à accomplir. Toutefois, l'espace des tâches envisagé concerne le domaine graphique uniquement. Comme pour les deux taxonomies précédentes, la caméra comme dispositif n'est pas considérée. De plus l'interaction bi-manuelle n'est pas explicitée. De plus la classification de Foley rend difficile la comparaison des dispositifs entre eux : comme le montre la Figure 5, un dispositif, qui peut se retrouver dans plusieurs sous-arbres, n'a pas une localisation unique dans l'espace de conception. Sur ce point, les propositions de Buxton et Mackinlay sont plus utiles.

1. 4 Conclusion sur les dispositifs

Nous avons vu la taxonomie de Buxton qui ne concerne que les dispositifs physiques continus. Les dispositifs qui ne sont pas actionnables par le geste instrumental comme des gestes reconnus par une caméra ne sont pas traités ainsi que tous les dispositifs bi-manuels.

Mackinlay constate des manques dans la taxonomie de Buxton et tente d'y remédier à l'aide de la description des dispositifs d'entrée au moyen d'une algèbre simplifiée. En introduisant des opérateurs de composition, Mackinlay nous invite alors à considérer les dispositifs bi-manuels comme la fusion de dispositifs (un pour chaque main) à l'image de la souris qui est la fusion de

deux dispositifs l'un contrôlant l'axe des x et l'autre considérant l'axe des y. Enfin, la taxonomie de Foley nous semble intéressante par l'introduction d'un niveau supplémentaire : la tâche réalisable avec le dispositif. Nous allons approfondir ce point dans le paragraphe suivant en considérant les modalités d'interaction et non les dispositifs pour la réalisation d'une tâche.

2 Modalité d'interaction et multimodalités

L'originalité de notre démarche est de considérer l'interaction bi-manuelle comme une interaction multimodale. Il s'agit alors d'observer dans quelle mesure l'interaction à deux mains vues comme deux modalités peut se décrire dans les espaces de conception de la multimodalité. Nous figeons d'abord la terminologie, puis nous étudions les espaces de conception existants.

Deux manières d'appréhender le problème : Soit ces notions sont abordés par rapport à l'individu, soit par rapport au système. Bien que notre approche soit centrée sur l'utilisateur, notre objectif est d'établir un espace de conception pour l'interaction bi-manuelle. Aussi notre point de vue est plus système.

Nous allons d'abord définir les termes modalité et multimodalité. Puis nous exposons trois espaces de conception qui sont complémentaires. Nous verrons d'abord celui proposé par Frolich, puis l'espace MSM. Enfin nous terminerons par les propriétés CARE et CARE étendues.

2. 1 Définition : modalité d'interaction et multimodalité

La variété des définitions du terme modalité, laisse entrevoir la difficulté d'identifier une définition consensuelle pour la multimodalité [Nigay 96]. Le terme modalité est tiraillé entre d'une part, les préoccupations des psychologues et des ergonomes et d'autres part, les considérations des concepteurs et informaticiens. Aussi pour nos travaux nous adoptons la définition de L. Nigay [Nigay 96] : une modalité est définie comme le couple <D, L> où :

- D désigne un dispositif physique (par exemple : la caméra + le doigt),
- L désigne un langage d'interaction (par exemple pour nos travaux : le langage d'interaction de manipulation de graphes par manipulation directe (Pan/Zoom/Rotation/Sélection).

Cette définition caractérise donc en particulier les échanges de l'utilisateur vers le système car permet de mettre en relation deux niveaux d'abstraction distincts que sont le niveau physique (dispositif) et le niveau logique (langage d'interaction). Les modalités sont utilisées pour réaliser une tâche qui est le troisième niveau d'abstraction comme le montre la Figure 6 issue de [Nigay 94]. Ce niveau tâche correspond au niveau tâche graphique de la taxonomie de Foley du paragraphe Chapitre III1. 3 L'utilisateur réalise des tâches en vue d'atteindre un but. Pour cela, il élabore un plan d'actions qui est une représentation mentale d'une suite de tâche à réaliser. Nous pouvons alors rajouter un quatrième niveau après le niveau tâche qui est celui du but à atteindre.

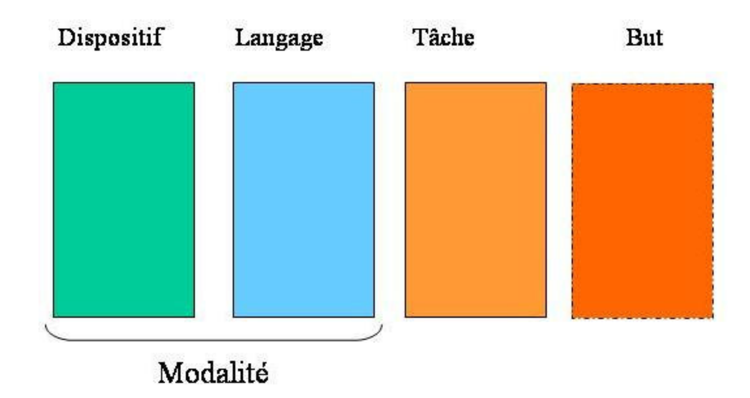


Figure 6 : Les quatre niveaux d'abstraction (dispositif, langage, tâche, but) défini dans [Nigay 94] et étendu avec [Dietrich 93]

La multimodalité en entrée comme en sortie se base sur notre définition d'une modalité (le couple <D, I>), le préfixe multi traduisant le nombre. L'étude de la multimodalité implique donc de considérer le nombre de dispositifs et le langage d'interaction distincts, constitutifs des passerelles logiques et physiques entre le système et l'utilisateur. La multimodalité revêt alors les trois aspects suivants :

- Multi-dispositif et mono-langage.
- Multi-dispositif et multi-langage.
- Mono-dispositif et multi-langage.

La multimodalité peut aussi bien être étudié d'un point de vue entrée (de l'utilisateur vers le système) ou en sortie (du système vers l'utilisateur). Comme nous nous intéressons à l'interaction bi-manuelle, nous aborderons seulement les formes de multimodalités en entrée.

2. 2 Espace de FROLICH

Cette espace [Frohlich 91] prend comme point de départ le modèle du processeur humain pour définir un espace de conception. Frolich distingue les entrées et les sorties. Il y a donc deux espaces de conception, nous considérerons seulement l'espace de conception d'entrée à la Figure 7. Cinq notions sont distinguées : le mode, le canal de communication, le canal de l'interface, le média et le style.

Le mode définit les deux métaphores d'interaction qui sont le langage et l'action.

Le canal de communication représente le sens humain mis en jeu. Le canal de l'interface est en correspondance directe avec le canal de communication ; il désigne la capacité sensorielle de l'interface permettant « d'écouter » le canal de communication de l'utilisateur. Par exemple, au canal de communication qu'est la voix, le canal de l'interface correspondant est l'audio.

Le média désigne le système représentationnel pour l'échange d'informations.

Le style correspond aux techniques d'interaction comme les champs à remplir ou les icônes.

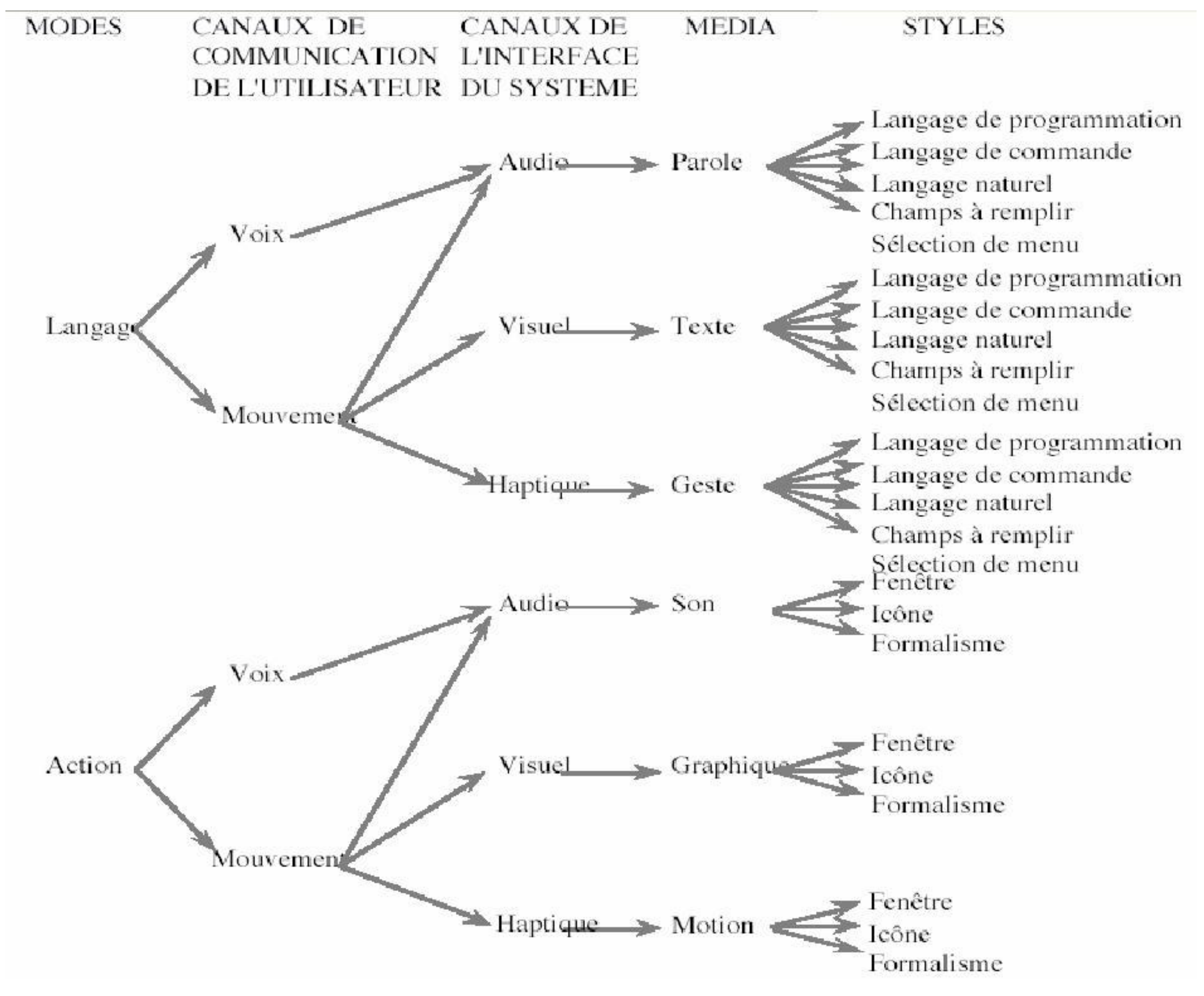


Figure 7 : Espace de conception des interfaces en entrée de Frolich

En accord avec notre terminologie d'une modalité, le mode et les canaux représentent le dispositif alors que le média et le style sont des affinements du langage d'interaction. Cet espace met en évidence que le sens humain constitue une caractéristique des dispositifs. Cependant, nous constatons que le canal de communication de l'utilisateur ne fait pas de distinction entre la main dominante et la main non dominante, distinction qui semble importante de souligner en conception.

2. 3 Espace MSM

L'espace MSM [Nigay 96] défini des axes pertinents pour la conception de systèmes multimodaux. Il propose les six axes reproduits à la Figure 8 comme cadre d'étude de la multimodalité.

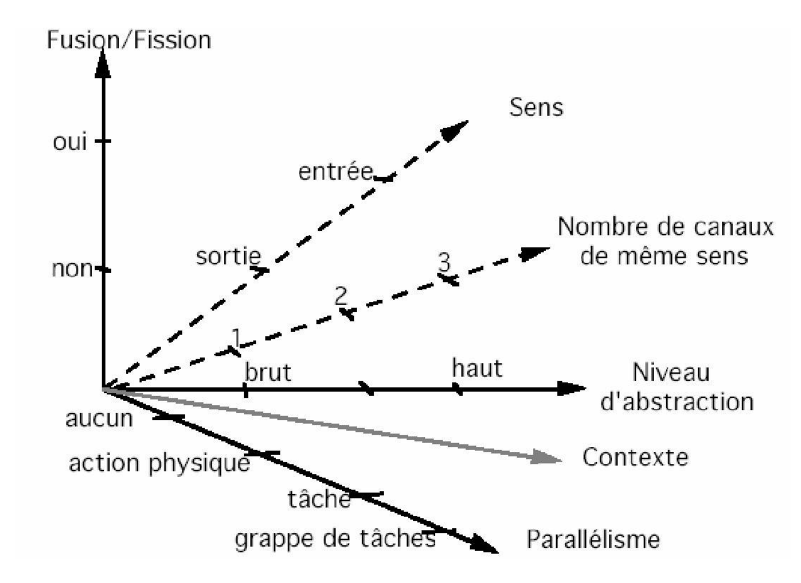


Figure 8 : L'espace de conception MSM ([Nigay 96])

Les deux dimensions "Sens" et "Nombre de canaux de même sens" ont trait à la notion de dispositif d'entrée et de sortie. Nous nous intéresserons ici seulement aux canaux d'entrée. Les autres dimensions caractérisent le degré de sophistication computationnelle des fonctions d'acquisition et de restitution du système : là encore, nous nous intéressons qu'à la fonction d'acquisition. Cette dernière se caractérise parles dimensions suivantes : niveaux d'abstraction, contexte, fusion et fission, granularité du parallélisme.

2. 3. 1. 1 Dispositif d'entrée

Les dispositifs d'entrée ou capteurs sont capables de recevoir des informations de types données sous le contrôle d'une capacité computationnelle ou processus. Cette définition met l'accent non pas sur l'aspect liaison, mais sur les intervenants de l'acte de communication : la source et le récipien.

2. 3. 1. 2 Fonction d'acquisition

L'information acquise par les dispositifs d'entrée est transformée par les processus internes du système. Cette suite d'opérations modélise la *fonction d'acquisition*. A chaque dispositif d'entrée, nous associons une fonction d'acquisition qui prolonge au cœur du système, les capacités de calcul du dispositif. Ce point correspond au mécanisme d'abstraction de la théorie de Mackinlay Chapitre III1. 2 .

Niveaux d'abstraction

La notion de niveau d'abstraction exprime le degré de transformations subies par les informations reçues par les dispositifs. Elle couvre également l'éventail des représentations que gère le système depuis les informations brutes (les signaux) jusqu'aux représentations symboliques (le sens). Une fonction d'acquisition se caractérise par son *pouvoir d'abstraction* des événements reçus du dispositif d'entrée auquel elle est associée.

Contexte

La capacité d'une fonction à abstraire peut dépendre de *variables contextuelles* ou *contexte*. Un contexte comprend un ensemble de variables d'état utilisées par les processus internes pour contrôler l'acquisition. Par exemple, dans les systèmes de traitement de texte, les saisies au clavier ou à la souris qui correspondent à des commandes sont comprises par le système tandis que ces mêmes saisies à destination du contenu des documents ne sont pas interprétées. Ainsi, le pouvoir d'abstraction d'un système est propre à chaque dispositif, mais pour un dispositif donné, ce pouvoir dépend du contexte.

Fusion et fission d'information

La *fusion* est la combinaison de plusieurs unités d'information pour former de nouvelles unités. La *fission* correspond au processus inverse. L'une et l'autre traduisent deux activités importantes des processus d'acquisition.

En acquisition, la *fusion* peut intervenir à un bas niveau d'abstraction entre des informations provenant de plusieurs dispositifs d'entrée. Elle peut aussi s'effectuer à un plus haut niveau d'abstraction pour des informations issues de différents contextes. Par exemple, l'expression orale "mets ça ici" nécessite la fusion de l'événement parole reçu via le microphone avec les événements souris qui désignent l'objet et le lieu dont il est question. Plus fréquente est la fusion de clics souris répartis sur une palette et une zone de dessin pour construire une figure géométrique. Ces informations issues du même dispositif (la souris) transitent selon des contextes différents pour être regroupés à un haut niveau d'abstraction [NIG 94].

La *fission* en acquisition traduit le besoin de décomposer une information issue d'un dispositif ou d'un contexte pour franchir un niveau d'abstraction. Par exemple, l'acte de parole "dessine un cercle dans une nouvelle fenêtre" fait référence à deux domaines de discours : les figures géométriques ("dessine un cercle") et l'interface homme-machine ("nouvelle fenêtre). Cette phrase dont le sens a pu être identifié à partir de données captées par un seul dispositif (le microphone) doit être décomposée en deux tâches élémentaires de haut niveau du système : "créer fenêtre" à destination du gestionnaire de fenêtre et "créer cercle" dans la nouvelle fenêtre à destination de l'éditeur graphique.

Parallélisme

Le parallélisme se manifeste à trois niveaux de granularité : action physique, tâche élémentaire, grappe de tâches.

Au niveau *physique*, le parallélisme en entrée autorise l'utilisateur à agir simultanément sur plusieurs dispositifs d'entrée comme dans l'exemple "mets ça ici".

Au niveau *tâche élémentaire*, le parallélisme en entrée autorise l'utilisateur à construire plusieurs commandes de front : en parallélisme vrai si le parallélisme est géré au niveau physique ou, en l'absence de parallélisme physique, de manière entrelacée (dialogue à plusieurs fils). Cet usage implique une certaine expertise de l'utilisateur.

Nous appelons *grappe de tâches*, un ensemble de tâches élémentaires permettant à l'utilisateur d'accomplir une tâche composée. Au niveau grappe de tâches, le parallélisme exprime les possibilités d'entrelacement entre des espaces de travail. Par exemple, l'utilisateur passe de l'édition d'un fichier teste à celle d'un dessin. Chaque espace d'édition correspond à une grappe de tâches.

2. 3. 1. 3 Parallélisme/séquentialité et fusion/fission

Le parallélisme (ou la séquentialité) couplé(e) au phénomène de fusion (ou de fission) offre un champ intéressant de réflexion présenté à la Figure 9. Comme le montre la ligne du centre de la Figure 9, la production séquentielle (case 3) ou simultanée (case 4) de deux (ou plusieurs) informations peut donner lieu à une fusion pour faire sens:

- Dans le premier cas, on parlera d'*utilisation alternée* de modalité : les modalités sont utilisées de manière séquentielle ou bien, le système impose une utilisation séquentielle.
- Dans le second cas, la fusion avec production simultanée (ou quasi simultanée) d'informations correspond à un *usage synergique* de plusieurs modalités. Cet usage est illustré par le paradigme du "mets ça là" [BOL 80].

Que l'usage soit synergique ou alterné, il y a émergence de sens par suite d'une fusion d'informations. Ce phénomène se produit lorsque des modalités entretiennent une relation de *complémentarité* : chaque modalité véhicule une partie d'information qui, prise isolément, est insuffisante pour faire sens.

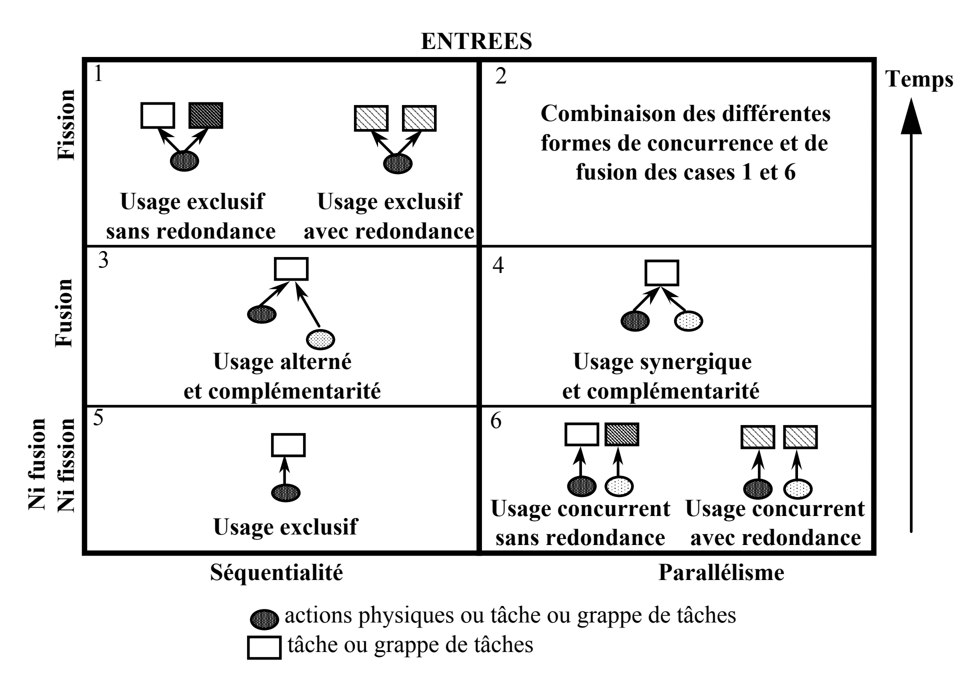


Figure 9 : Les différents cas d'usage en entrée de modalités (séquentielle/parallèle ;fusion/fission) [Nigay 96]

La ligne du bas de la Figure 9, correspond à l'absence de fusion et de fission. On parle *d'usage exclusif* lorsque l'utilisateur emploie (ou lorsque le système impose) au plus une modalité à la fois. Dans la case 5, on n'observe ni fusion, ni parallélisme. Par exemple un système uniquement vocal sur téléphone n'autorise l'usage que d'une seule modalité de façon exclusive, la parole. Mais comme le montre la case 1, l'usage exclusif d'une modalité peut amener le système à pratiquer une fission. Le résultat de cette opération conduit à plusieurs informations distinctes comme dans l'exemple de l'énoncé oral ("Dessine un cercle dans une nouvelle fenêtre").

La production simultanée de plusieurs informations ne conduit pas nécessairement à une fusion. Chaque unité d'information peut faire sens. Il s'agit alors d'un *usage concurrent* (voir case 6). L'utilisateur mène alors de front deux tâches distinctes (comportement expert). La

redondance est un cas particulier de concurrence lorsque les deux informations produites en parallèle couvrent le même sens

La complémentarité et la redondance caractérisent l'usage de plusieurs modalités. Orthogonale à la notion de multimodalité, vient la notion de choix effectué par l'utilisateur, notion traitée par les propriétés CARE.

2. 4 Propriétés CARE

Les propriétés CARE sont au nombre de quatre et définissent le lien entre les modalités, dans le contexte d'une interaction multimodale. C, A, R, E sont les initiales de quatre propriétés : Complémentarité, Assignation, Redondance et Equivalence. Dans [Coutaz 95], les auteurs montrent leur utilité pour classer et surtout évaluer les systèmes multimodaux, par l'analyse des propriétés CARE du point de vue système (S-CARE, "système CARE") et utilisateur (U-CARE "utilisateur-CARE).

Les propriétés CARE s'appliquent pour le cas de modalités d'entrée et de sortie. Dans le cadre de notre étude de l'interaction bi-manuelle, nous ne considérons que les modalités d'entrée.

- L'*assignation* exprime l'obligation, pour un état de départ et d'arrivée fixés, d'utiliser une modalité. Cette propriété exprime donc l'absence de choix pour réaliser une tâche, consistant à passer d'un état de départ à un état d'arrivée. Ceci n'est pas forcément un point négatif pour l'interaction. L'assignation dénote cependant un manque évident de souplesse.
- L'*équivalence* de modalités se définit pour un état de départ et d'arrivée fixés : l'équivalence d'un ensemble de modalités est vérifiée si chaque modalité permet d'atteindre l'état d'arrivée depuis l'état de départ en une seule étape. Deux modalités sont équivalentes pour la réalisation d'une tâche. L'utilisateur a alors le choix des modalités pour réaliser sa tâche.
- La *redondance* dénote l'utilisation séquentielle ou parallèle (Espace MSM présenté ci-dessus) de plusieurs modalités équivalentes. En entrée, la redondance d'information en provenance de l'utilisateur implique la prise en compte d'une seule des modalités par le système, l'autre pouvant éventuellement contribuer à désambiguïser l'expression obtenue.
- La *complémentarité* entre les modalités d'un ensemble M exprime le fait que pour passer d'un état à un autre, il faille utiliser toutes les modalités de l'ensemble M. Cela signifie qu'aucune des modalités de l'ensemble M ne suffise à elle seule pour passer de l'état de départ à l'état d'arrivée. Toutefois, il n'est pas exclu qu'une modalité non contenue dans M puisse permettre cette transition. Comme pour la redondance, l'utilisation complémentaire de modalités peut être parallèle ou séquentielle.

2. 5 Propriétés CARE étendues [Vernier 01]

Dans [Vernier 01] les propriétés CARE sont affinées avec la mise en relation de cinq schémas de composition avec cinq aspects de composition qui sont le temps, l'espace,

l'articulation des dispositifs, la syntaxe des langages représentationnels et la sémantique de l'information véhiculée. Dans les aspects outre les dimensions spatio-temporelles de la composition de deux modalités, nous retrouvons les niveaux dispositif (articulatoire) et langage (syntaxique) de la définition d'une modalité tandis que l'aspect sémantique caractérise la complémentarité et la redondance de CARE. La Figure 10 illustre cette mise en correspondance. Tirés des travaux d'Allen [Allen 83] sur l'aspect temporel, les cinq schémas de composition illustrent la composition de modalités éloignées (1ère colonne), la composition de modalités avec un point de contact (2ème colonne), la composition de modalités avec une intersection non vide (3ème colonne), la composition de modalités dont l'une est plus étendue et englobe l'autre (4ème colonne) et la composition de modalités de même étendue (5ème colonne).

Cet espace de classification affine les deux propriétés CARE concernant la fusion, c'est-à-dire la complémentarité et la redondance (aspect sémantique de la composition, dernière ligne). Ainsi, nous remarquons que les deux premières colonnes sont relatives à la complémentarité alors que les deux dernières correspondent aux différents cas de redondance (respectivement redondance partielle et redondance totale). La colonne du milieu définit la relation qu'il peut exister entre ces deux propriétés (complémentarité et redondance).

Aspects de composition	Schémas de composition				
Composition					
Temporelle	Anachronique	Séquentielle	Concomitante	Coïncidente	Parallèle / Simultanée
Spatiale	Disjointe	Adjacente	Intersectée	Imbriquée	Recouvrance
Articulatoire	Indépendance	Fissionnée	Fissionnée + Dupliquée	Partiellement Dupliquée	Dupliquée
Syntaxique	Différente	Complétion	Divergence	Extension	Jumelage
Sémantique	Concurrente	Complé-mentaire	Complémentaire + Redondante	Partiellement Redondante	Totalement Redondante

Figure 10 : CARE étendues. Application des schémas de composition aux cinq aspects de composition [Vernier 01].

2. 6 Conclusion sur la multimodalité

Dans le cadre de notre étude sur l'interaction à deux mains, nous avons étudié plusieurs espaces de conception de la multimodalité en focalisant sur les modalités d'entrée. De cet état de l'art,

- Nous concluons que :
  - La distinction entre les deux mains n'a jamais été explicitée dans les espaces sur la multimodalité. En cela notre approche est originale.
  - L'interaction bi-manuelle correspond à une interaction multimodale multi-dispositif, mais qui peut être mono ou multi-langage.
  - L'interaction bi-manuelle s'inscrit dans la métaphore d'interaction Action.

- Nous retenons pour nos travaux :
    - La notion de parallélisme des actions affinée grâce aux relations d'Allen.
    - La fusion (redondance ou complémentarité) des informations en provenance des 2 mains.

L'espace de composition de la Figure 10 nous semble particulièrement pertinent pour notre étude pour décrire finement la composition qui s'opère entre les modalités associées à chaque main. En particulier les relations d'Allen peuvent nous aider à caractériser précisément les actions des deux mains dans le temps. Nous allons d'ailleurs reprendre les résultats d'Allen dans la partie suivante pour étendre l'un des deux espaces de conception dédié à l'interaction bi-manuelle.

3 Espace de conception des interactions bi-manuelles

L'espace de conception de Buxton [BUX 05] dédié à l'interaction bi-manuelle s'appuie sur les travaux de Guiard [GUI 87] présentés au chapitre II. Cet espace caractérise les types d'actions effectuées par les deux mains. La nature de l'action ou le « langage » est directement liée aux dispositifs d'entrée disponibles. En conception, le langage choisi devrait influencer le choix des dispositifs d'entrée. En pratique, les dispositifs d'entrée sont souvent fixés avant les langages et dictent les limites de langage d'interaction par leurs propriétés physiques. Selon Buxton, le langage d'actions le plus simple est défini par deux classes : action discrète et action continue. Une classe intermédiaire regroupant des actions qualifiées de composées, est ensuite identifiée. Outre les types d'actions, l'espace considère aussi la temporalité des actions.

Dans ce paragraphe, nous exposons cet espace de conception en considérant d'abord le langage d'action puis la temporalité des actions.

3. 1 Action discrète, Action continue

Le langage d'actions comprend deux classes élémentaires d'actions : action discrète et action continue.

- Action discrète : les actions discrètes sont des actions qui impliquent le déclenchement discret des événements, typiquement l'action d'appuyer sur un bouton. Cela comprend donc la saisie au clavier AZERTY ou l'utilisation de touches spéciales, mais également les actions de pression sur des boutons d'un dispositif continu tel qu'une souris.
- Action continue: les actions continues sont les actions qui impliquent le contrôle continu d'un ou plusieurs degrés de liberté. Ceci inclut les différentes interactions dans les interfaces graphiques traditionnelles, comme pointer et déplacer. Ces actions sont généralement effectuées à l'aide d'une souris, trackball ou joystick.

La caractéristique Discrète/Continue concerne la nature de l'action elle-même et non le résultat de l'action. Par exemple, dans les jeux, le mouvement des personnages est continu alors que les actions sont discrètes en enfonçant les touches d'un clavier. Réciproquement, un dispositif comme un bouton cranté (Figure 4), sur lequel les actions sont continues, est souvent utilisé pour saisir des données typiquement considérées comme "discrètes". En déclinant la caractéristique Discrète/Continue au cas de l'interaction bi-manuelle, nous obtenons le Tableau 1. Dans ce tableau, nous notons le cas Continue/Discrète qui est qualifié d'inadapté. En effet, même si les résultats de Y. Guiard identifient une complémentarité des deux mains, ils ne nient pas la supériorité de la main dominante. L'action continue étant considérée comme plus difficile qu'une action discrète, le cas où l'action la plus difficile est affectée à la main non dominante est donc inadapté.

Main non dominante	Main dominante	exemple
Discrète	Discrète	Taper sur un clavier AZERTY
Discrète	Continue	Utiliser une interface graphique avec une souris dans la main dominante et des touches spéciales dans la main non dominante
Continue	Discrète	**Inadapté**

Continue	Continue	Etirer une ligne à chaque extrémité, utilisant une trackball dans la main non dominante et la souris dans la main dominante

Tableau 1 : Classification des différents formes d'actions bi-manuelles en osidérant les actions comme étant discrètes et continues [Buxton 04]

Dans l'espace de Buxton, une troisième classe d'actions dite composée est introduite. Elle permet de distinguer des actions purement continues de celles qui incluent également une action discrète à l'aide du même dispositif. Ainsi les actions composées désignent des actions qui entrelacent des actions continues et des actions discrètes. La sélection par pointer-cliquer présente dans la plupart des interfaces graphiques ou GUI (*Graphical User Interface*) est un exemple d'action composée. En utilisant cette nouvelle classe d'actions, nous pouvons préciser la catégorisation des différentes formes d'actions bi-manuelles. Le tableau 2 présente cette catégorisation qui étend celle du tableau 1. Comme précédemment, certains cas sont inadaptés car les actions composées sont considérées comme plus difficiles que les actions continues ou discrètes.

Main non dominante	**Main dominante**	exemple
Discrète	Discrète	Taper sur un clavier AZERTY
Discrète	Continue	Positionner un curseur avec un joystick dans la main dominante et utiliser des touches spéciales avec la main non dominante.
Discrète	Composée	Pointer et cliquer avec une souris dans la main dominante, utiliser des touches spéciales avec la main non dominante.
Continue	Discrète	***Inadapté***
Continue	Continue	Etirer une ligne à chaque extrémité, utilisant une trackball dans la main non dominante et la souris dans la main dominante
Continue	Composée	Scroller avec une trackball dans la main non dominante et pointer et cliquer avec une souris dans la main dominante.
Composée	Discrète	***Inadapté***
Composée	Continue	***Inadapté***
Composée	Composée	Tâche complexe de dessin (T3 [Fitzmaurice 97])

Tableau 2 Classification complexe des différentes formes d'action bi-manuelles [Buxton 04]

Outre les types d'actions effectuées par les deux mains, l'espace considère aussi la temporalité des actions.

3. 2 Temporalité des actions

Selon Buxton, l'autre manière de décrire l'interaction bi-manuelle consiste à considérer les relations temporelles des actions effectuées par les deux mains. L'objectif est ici double : classer les différentes actions bi-manuelles, mais aussi comparer les coûts et les bénéfices de l'interaction à deux mains par rapport à l'interaction à une main.

3. 2. 1 Temporalité dans l'interaction à une main

Supposons deux sous-tâches (A) et (B). Dans un éditeur de dessin, la sous-tâche A peut être par exemple le fait de dessiner avec l'outil pinceau, alors que la sous-tâche B consiste à choisir des propriétés dans la palette. Pour réaliser sa tâche de dessin, l'utilisateur doit alterner la sous-tâche (A) et la sous-tâche (B). Entre chaque sous-tâche, il y a un temps de transition lié à un coût "cognitif " mais aussi articulatoire, explicité à la Figure 11. Dans notre exemple, le temps de transition correspond au temps pour positionner son regard sur la palette, puis le temps pour sélectionner une propriété de la palette et enfin le temps pour revenir dans le contexte graphique du dessin en cours d'édition. Un exemple similaire réside dans le défilement d'un texte en cours d'édition.

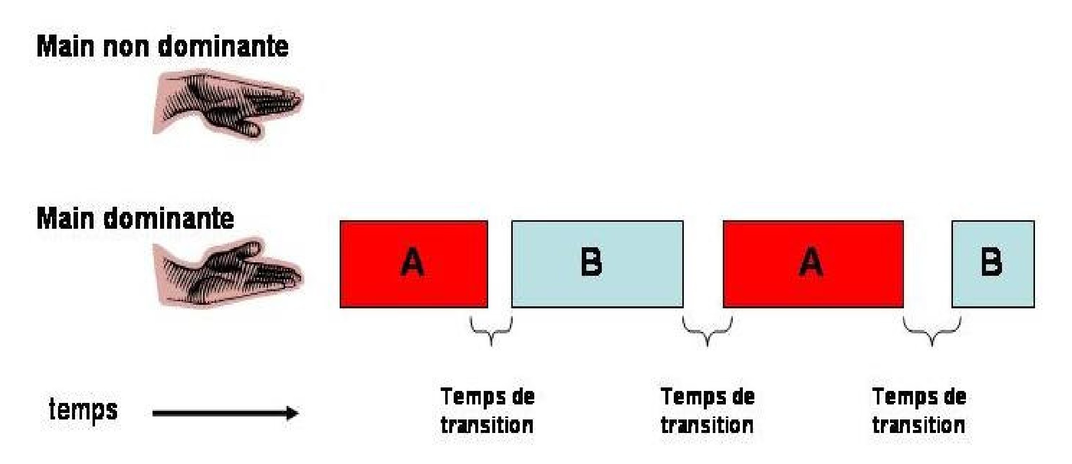


Figure 11 : Réaliser les sous-tâches A et B en utilisant une approche à une main [Buxton 05]

3. 2. 2 Temporalité dans l'interaction deux mains

Considérons maintenant l'interaction à deux mains, avec un dispositif d'entrée pour chaque main et chaque sous-tâche. Comme chaque main a maintenant l'outil approprié, et qu'il peut être à la bonne position pour chaque sous-tâche, cette approche élimine les temps de transition associés à l'acquisition du dispositif et de changement d'attention visuelle sur le dispositif à utiliser. Par conséquent, même si les deux mains sont utilisées de manière strictement séquentielle, et même s'il reste encore un temps de transition "cognitif ", il y a une amélioration significative dans l'exécution, comme le montre la Figure 12. Ce résultat a été montré expérimentalement [Buxton 86] dans le cas où l'utilisateur faisait défiler un document avec une surface tactile de la main non dominante tout en sélectionnant le texte avec une souris de la main dominante. De plus les sujets qui faisaient du parallélisme étaient les plus rapides. Ceci implique évidemment que les sous-tâches puissent se faire en parallèle. Dans le cas idéal de la Figure 13, le parallélisme est maximisé. Par exemple, une évaluation expérimentale de l'interaction à deux mains avec des "toolGlasses" [Bier 94] montre qu'en moyenne, 83% du temps se déroule en activités parallèles.

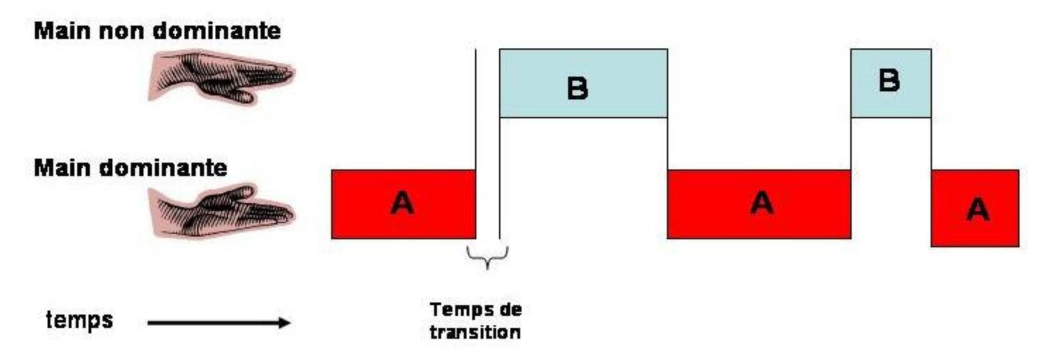


Figure 12 : Réaliser les sous-tâches A et B en utilisant une approche à deux mains [Buxton 04]

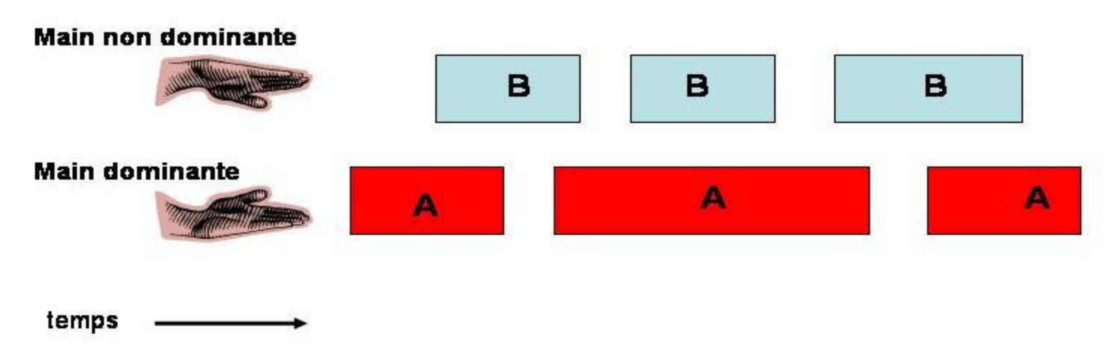


Figure 13 : Réaliser les sous-tâches A et B en introduisant du parallèlisme [Buxton 04]

L'efficacité de l'interaction à deux mains par rapport à celle à une main est néanmoins dépendante des sous-tâches considérées. Deux sous-tâches peuvent même avoir un coût cognitif si élevé que la technique à deux mains est encore plus lente qu'une technique à une main. Dans [Kabbash 94] cette assertion a été prouvée expérimentalement dans le cas où l'utilisateur devait colorier des points et des segments. Ce coût "cognitif" de changement de sous-tâches était tel que les auteurs ont nié les avantages d'utiliser les deux mains dans l'interaction. A la Figure 14, nous illustrons une interaction bi-manuelle avec un coût cognitif élevé.

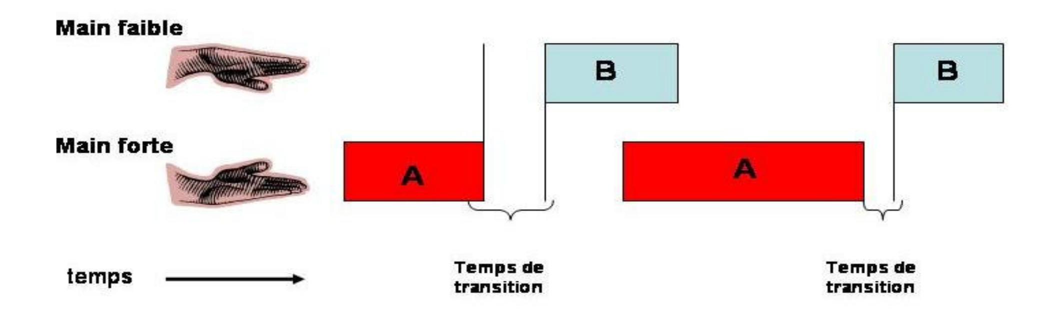


Figure 14 : Réaliser les sous-tâches A et B avec un coût cognitif élevé [Buxton 04]

3. 3 Conclusion sur l'interaction bi-manuelle

L'espace de conception de Buxton analyse l'interaction bi-manuelle selon deux axes : le type d'actions et la temporalité des actions. Trois types d'actions sont identifiés : action discrète,

continue et composée (classée selon leur complexité). Pour les aspects temporels de l'interaction bi-manuelle, l'espace souligne deux cas : actions séquentielles et actions en parallèle (ou recouvrement). Nous proposons dans le chapitre suivant d'étendre le modèle temporel de Buxton afin d'affiner la notion de parallélisme en introduisant les relations d'Allen.

Cet espace permet de caractériser l'interaction bi-manuelle mais ne fournit pas de réponses à des questions principales comme : Quelles sont les sous-tâches qui peuvent être effectuées en parallèle ? Quelle forme de dépendance y a-t-il entre les sous-tâches ? Existe-t-il un lien entre la dépendance des sous-tâches et la dépendance des mains comme le propose Y. Guiard ?

4 Conclusion sur les espaces de conception

Après avoir étudié l'interaction bi-manuelle d'un point de vue humain au chapitre 1, nous avons consacré ce chapitre à la conception de techniques d'interaction bi-manuelle. Pour cela nous avons d'abord étudié les espaces taxinomiques des dispositifs. Nous notons l'approche de Mackinlay qui est la seule à permettre de considérer le dispositif bi-manuel, composé des deux dispositifs manipulés par les deux mains, grâce à son opérateur de fusion de dispositifs. Nous avons ensuite analysé les résultats dédiés à l'interaction multimodale. En effet, l'originalité de notre approche est de considérer une modalité associée à chaque main. Aucun espace de conception de la multimodalité ne considère explicitement l'interaction à deux mains. Nous retenons de ces travaux l'usage combiné de plusieurs modalités et les relations temporelles dans l'usage des modalités. Nous avons enfin exposé le seul espace de conception qui soit dédié à l'interaction bi-manuelle. Trois types d'actions sont identifiés et les aspects temporels dans la réalisation des sous-tâches sont étudiés. L'axe temps est donc commun aux espaces de conception dédiés à l'interaction multimodale et l'espace de conception pour l'interaction bi-manuelle. Ce point est développé dans le chapitre suivant où nous présentons notre contribution, un nouvel espace de conception, nommé 2M, qui prend ses fondements dans les études exposées dans ce chapitre.

Chapitre IV Nouvel espace de conception bi-manuel 2M

Dans ce chapitre consacré à la conception de l'interaction bi-manuelle, nous présentons un nouvel espace de conception, noté 2M. Cet espace organise dans un canevas unificateur des résultats issus d'études sur l'interaction multimodale et bi-manuelle que nous avons exposés au chapitre précédent.

Nous organisons l'exposé de notre espace 2M comme suit : au paragraphe 1, nous enrichissons les travaux sur l'interaction multimodale pour prendre en compte le rôle des deux mains et définissons ainsi des niveaux d'abstraction pour l'étude de l'interaction bi-manuelle. Nous étudions ensuite l'interaction à deux mains au regard de la multimodalité (une modalité par main) en affinant d'abord l'axe temporel de l'espace de conception de Buxton [Buxton 05]. Le paragraphe 3 est ensuite consacré à présenter notre nouvel espace 2M qui permet de prendre en compte les aspects temporels mais aussi de dépendance de la multimodalité lors de la conception de techniques d'interaction bi-manuelle. Les axes de l'espace 2M exposés, nous analysons ensuite le pouvoir descriptif et taxinomique de l'espace 2M en considérant de nombreux systèmes bi-manuels puis nous étudions son pouvoir génératif et sa capacité à guider la conception de formes d'interaction bi-manuelle.

1 Multimodalité et interaction bi-manuelle : niveaux d'abstraction

Basé sur la définition d'une modalité comme un couple (dispositif, langage) [Nigay 94], trois niveaux d'abstraction décrits au chapitre précédent permettent de raisonner sur l'interaction multimodale : dispositif, langage et tâche. Nous avions rajouté un quatrième niveau d'abstraction, le but [Dietrich 93], que cherche à atteindre l'utilisateur par la réalisation de tâches selon plusieurs modalités. Pour la prise en compte de l'interaction bi-manuelle, nous ajoutons un cinquième niveau, le canal gestuel comme présenté à la Figure 15. Le canal gestuel [Cadoz 94] est défini au chapitre II et correspond au canal de communication "mouvement" du mode "action" dans l'espace de Frohlich. Dans le cadre de notre étude, nous réduisons le canal gestuel au cas de gestes avec la main dominante et non dominante. L'importance de la distinction entre les deux mains au sein du canal gestuel est soulignée par les travaux en psychologie de Y. Guiard [Guiard 97].

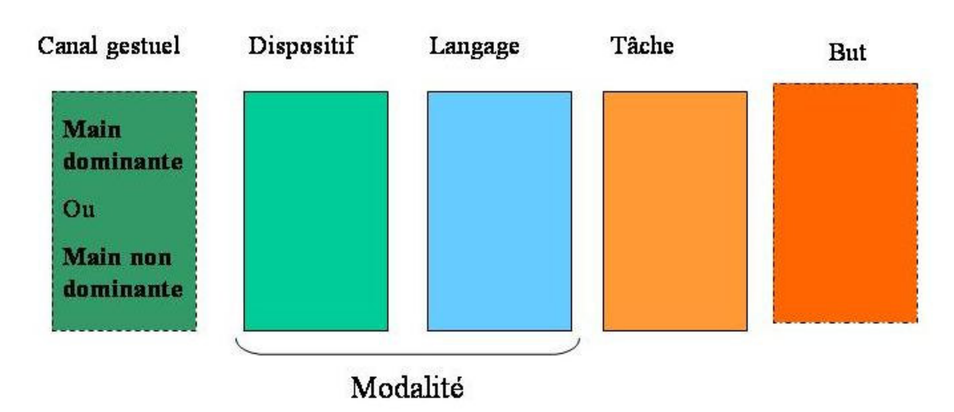


Figure 15 : Représentation des cinq niveaux d'abstraction pour l'étude de la multimodalité tirée du schéma de [Nigay 94] complété par [diestrich 93] et [Cadoz 94]

L'ajout du "canal gestuel" nous permet d'étudier chaque niveau d'abstraction au regard des gestes de la main dominante et de la main non dominante, comme le montre la Figure 16. Ainsi au premier niveau, il convient de caractériser le dispositif tenu par chaque main, et les espaces taxinomiques dédiés aux dispositifs exposés au chapitre précédent sont directement applicables ici. Au niveau suivant, l'importance de l'étude du langage d'interaction associé à chaque main a été soulignée par les travaux de Buxton. Une caractéristique immédiate du langage réside donc dans la distinction entre action discrète/continue/composée. Rappelons que cette caractéristique ne concerne pas le résultat de l'action c'est-à-dire le niveau tâche mais bien le langage qui doit abstraire des événements reçus du dispositif selon que l'action est considérée comme discrète/continue/composée. D'autres caractéristiques du langage issues de travaux en multimodalité sont à prendre en compte comme son caractère arbitraire ou non. Enfin nous soulignons que les langages associés aux deux mains peuvent être différents (multi-dispositif/multi-langage), ou similaires (multi-dispositif/mono-langage). Au niveau supérieur, tâche, il convient d'étudier les tâches associées à chaque main et les travaux en psychologie de Y. Guiard proposent un cadre d'étude de la répartition des tâches entre chaque main, comme le fait que la main non dominante fixe le contexte dans lequel la main dominante agit. Enfin le dernier niveau, but, nous permet de distinguer le cas où l'utilisateur utilise ces deux mains pour réaliser un but unique selon une décomposition hiérarchique des tâches à réaliser, de celui où l'utilisateur exploite les deux mains pour mener des activités totalement indépendantes. La distinction que nous faisons entre tâche et but n'est pas aussi simple : ainsi dans [NOR 92] un but est défini comme un état du système que l'utilisateur souhaite atteindre tandis qu'une tâche est un but que l'utilisateur vise à atteindre assorti d'une procédure (ou plan) qui décrit les moyens pour atteindre ce but. Ce plan consiste en une décomposition hiérarchique en sous-tâches. Ces définitions soulignent donc que la frontière entre tâche et but n'est pas aussi évidente que le présente notre Figure 15, mais qu'il s'agit plus d'un continuum des tâches élémentaires issues d'actions physiques jusqu'à un but. Aussi lors de l'étude de l'interaction bi-manuelle selon nos niveaux d'abstraction, l'enjeu est de ne pas mélanger au sein d'un même niveau des concepts de granularité différente. Nous constatons cette difficulté par exemple lors de l'étude des relations temporelles entre les actions des deux mains qui, selon le niveau d'abstraction considéré pour l'étude, ne seront pas les mêmes. Les aspects temporels de l'activité bi-manuelle sont étudiés au paragraphe suivant.

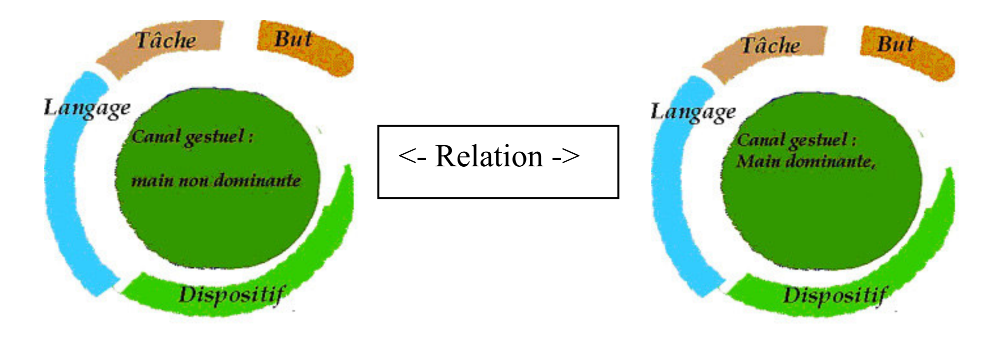


Figure 16 : Caractérisation des gestes de chaque main : des relations entretenues entre ces caractéristiques émerge une interaction bi-manuelle ou non.

2 Interaction bi-manuelle et multimodalité : aspects temporels

Les relations temporelles des sous-tâches affectées à chaque main ont été étudiées dans [Buxton 05] dans le cadre de l'interaction bi-manuelle au chapitre précédent. Les sous-tâches sont effectuées soit de façon séquentielle ou soit parallèle :

- Dans le cas de sous-tâches séquentielles, un temps de transition est éventuellement présent. Celui-ci est dû à un coût cognitif et ne concerne pas le temps de prise en main du dispositif comme dans le changement de dispositifs lors d'interaction mono-manuelle (opérateur "Homing" dans KEYSTROKE [Card 83]). Ce temps de transition est d'autant plus petit que les dispositifs sont en adéquation avec les sous-tâches à réaliser ou que l'utilisateur est expert.
- Le cas de sous-tâches parallèles souligne l'apport de l'interaction bi-manuelle sur l'efficacité de l'interaction. Ainsi des sous-tâches qui étaient initialement prévues pour être effectuées séquentiellement, peuvent être réalisées de manière parallèle si l'utilisateur est un expert. Par exemple, dans [Bourgeois 01], il est montré qu'il est possible de zoomer en même temps que de se déplacer.

Nous affinons les deux relations temporelles identifiées (séquentiel et parallèle) avec les relations d'Allen [Allen 83] de la Figure 17. Nous retenons que cinq relations temporelles des sept d'Allen, en ne considérant que celles utilisées comme des schémas de composition de modalités dans l'espace pour la multimodalité [Vernier 01] (Figure 10 du chapitre III). Au lieu de deux cas (séquentiel et parallèle), nous obtenons donc cinq cas. Pour chaque cas, nous reprenons la terminologie de la Figure 10 du chapitre III pour l'aspect temporel et nous le schématisons en nous inspirant des schémas (comme celui de la Figure 11 du chapitre III) de l'espace de Buxton [Buxton 05]).

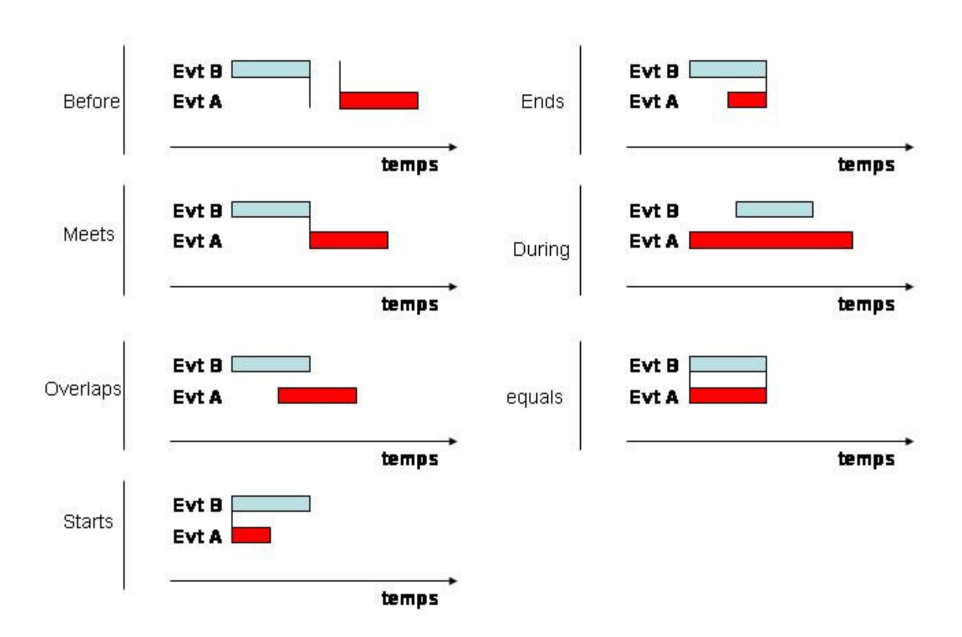


Figure 17 : Les sept relations d'Allen [Allen 83]

Anachronique ou séquentielle : Lorsque deux sous-tâches sont réalisées de manière séquentielle avec un temps de transition comme le montre la partie gauche de la Figure 18, elles

sont réalisées de façon anachronique. Si le temps de transition donc le coût cognitif tend à diminuer vers 0, la réalisation des sous-tâches se rapproche du cas séquentiel comme le montre la partie droite de la Figure 18. Aussi nous regroupons les deux cas anachronisme et séquentialité en un seul cas que nous notons séquence.

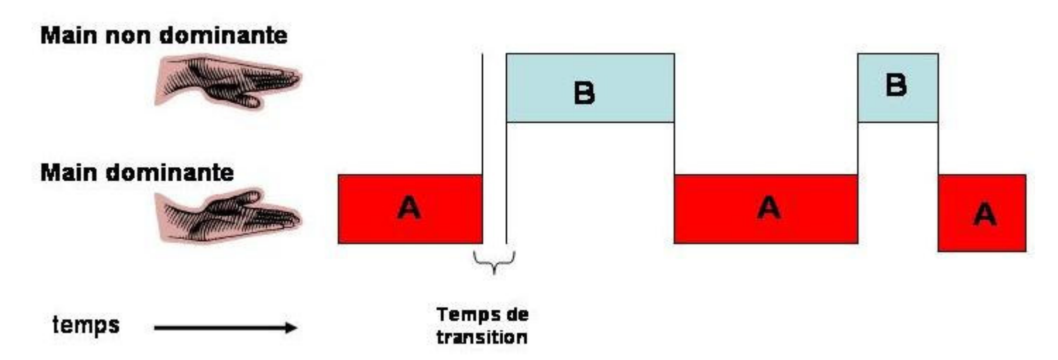


Figure 18 : Séquence de sous-tâches avec un temps de transition (partie gauche) et sans temps de transition (partie droite). Exemple : sous-tâche B = choix de la couleur dans une palette et sous-tâche A = coloriage de la couleur courante.

Concomitante : Les deux sous-tâches sont réalisées séquentiellement avec un recouvrement (chevauchent des sous-tâches) comme le schématise la Figure 19. Avant d'avoir terminé une sous-tâche, l'utilisateur expert commence déjà l'autre sous-tâche.

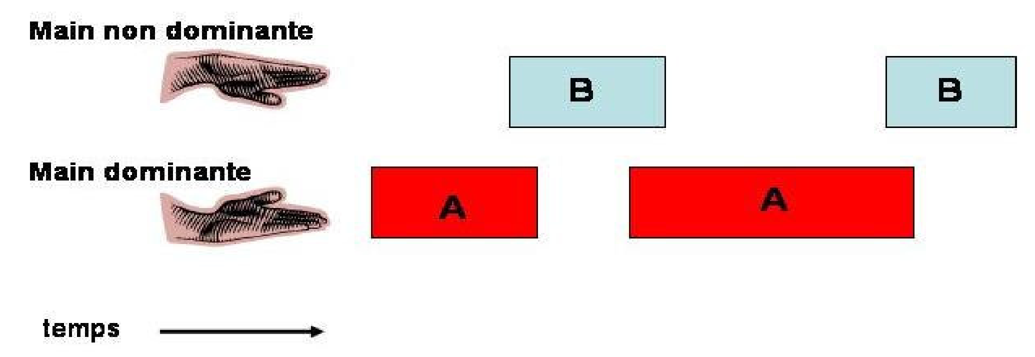


Figure 19 : Concomitance (chevauchement) des sous-tâches A et B. Par exemple les sous-tâche B = défilement et sous-tâche A = sélectionner pour un utilisateur expert.

Coïncidence : Une sous-tâche est incluse dans l'autre relation *During*. La Figure 20 présente le cas où la sous-tâche affectée à la main non dominante est incluse dans celle de la main-dominante. Cela correspond par exemple à l'utilisation de touches spéciales d'un clavier avec la main non dominante tout en manipulant la souris de la main dominante.

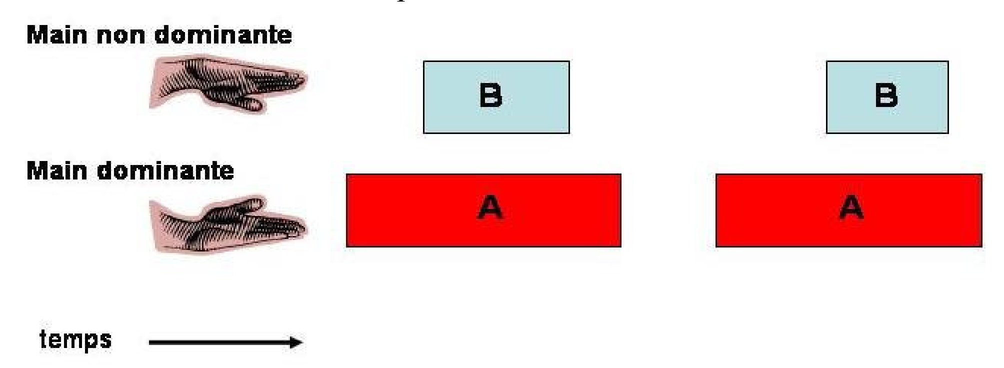


Figure 20 : Coïncidence (inclusion) des tâches A et B dans l'interaction à deux mains. Par exemple sous-tâche B = enfoncement d'une touche spéciale au clavier et sous-tâche A = sélection multiple avec la souris.

Parallèle : Les tâches sont executés en même temps durant toute la durée comme le schématise la Figure 21. Nous pouvons prendre l'exemple du char, ou pour avancer, il faut que chaque main appuie sur les accélérateurs durant toute la phase de déplacement.

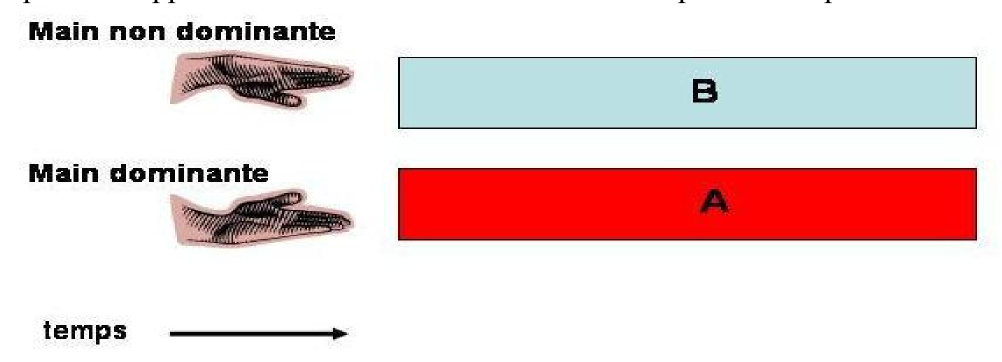


Figure 21 : Parallélisme des sous-tâches A et B dans l'interaction à deux mains. Par exemple pour faire avancer en ligne droite un char dans un jeu de réalité virtuelle, sous-tâche B = appui de l'accélérateur pour le contrôle de la vitesse d'une chenille d'un char et sous-tâche A = appui de l'accélrateur pour le contrôle de la vitesse de l'autre chenille d'un char.

En conclusion, nous avons étendu les deux relations temporelles de l'espace de Buxton [Buxton 05] à quatre relations en exploitant des résultats issus de l'étude de la composition de modalités [Vernier 01]. Cet affinement permet par exemple de distinguer le cas de la sélection multiple où la touche spéciale du clavier reste enfoncée le temps des sélections avec la souris, du cas du défilement d'un texte avec une trackball tout en sélectionnant un mot avec la souris. Avec uniquement les deux relations (séquentiel et parallèle) [Buxton 05], ces deux cas d'usage des deux mains pourtant différents ne sont pas distingués et tous deux décrits comme des sous-tâches en parallèle.

Nous constatons que les espaces de conception de l'interaction multimodale et celui de l'interaction bi-manuelle partagent la dimension temporelle. Les deux autres dimensions, respectivement dépendance pour la multimodalité et nature de l'action pour l'interaction bi-manuelle sont distinctes. Ces constats sont à l'origine de notre espace de conception 2M.

3 Nouvel espace de conception de l'interaction bi-manuelle : 2M

Comme nous l'avons expliqué dans le chapitre précédent, l'interaction bi-manuelle, comme la multimodalité peut s'avérer moins utilisable que l'interaction à une seule main. Dans ce contexte, il convient de caractériser finement l'interaction bi-manuelle afin de mieux comprendre et concevoir des techniques d'interaction bi-manuelle utilisables. Notre espace de conception, noté 2M, vise cet objectif en organisant les aspects de conception de l'interaction bi-manuelle au sein d'un canevas cohérent et unificateur.

3. 1 Apports d'un espace de conception

Comme tout espace de conception, l'apport de notre espace 2M est triple :

- Classification des techniques d'interaction bi-manuelle existantes. En vue d'une meilleure compréhension de l'ensemble des possibilités, l'espace 2M doit permettre de classer les systèmes bi-manuels, par exemple de regrouper ceux aux caractéristiques communes. L'espace 2M doit permettre également de situer les futurs systèmes bi-manuels par rapport à l'existant.
- Conception de nouvelles formes d'interaction bi-manuelle. L'espace doit permettre aux concepteurs de considérer de façon systématique les paramètres de conception. De plus par l'identification de zones de l'espace non encore explorées, l'espace doit favoriser la conception de nouvelles formes d'interaction.
- Comparaison/Evaluation de l'utilisabilité des techniques d'interaction bi-manuelle, en identifiant par exemple que certains points de notre espace correspondent à des techniques moins utilisables que d'autres. Il est alors possible d'envisager un guide de conception pour l'interaction bi-manuelle sur la base de l'espace 2M.

3. 2 Dimensions de l'espace 2M

L'originalité de notre approche est de combiner deux domaines qui sont la multimodalité et l'interaction bi-manuelle. Nous constatons que l'espace de conception de l'interaction multimodale et celui de l'interaction bi-manuelle partage la dimension temporelle. Les deux autres dimensions, respectivement dépendance pour la multimodalité et nature de l'action pour l'interaction bi-manuelle sont distinctes. Nous obtenons donc trois dimensions qui définissent notre espace 2M comme le montre la Figure 22:

- La temporalité : elle traduit les relations temporelles entre les sous-tâches affectées à chaque main. Nous utilisons les quatre relations que nous avons définies au paragraphe 2 : séquentielle, concomitante, coïncidente et parallèle.
- La nature des actions : cette dimension considère des couples (Main Non-Dominante, Main Dominante) notés (MND, MD) qui reprennent les différentes actions définies dans [Buxton 05]. Le tableau 2 du chapitre III présente les six couples possibles en excluant trois cas qualifiés d'inadaptés. Nous obtenons donc un axe "nature des actions" à six valeurs discrètes, par exemple (MND-Discrète, MD-Continu).
- La dépendance : Elle traduit si les sous-tâches sont dépendantes ou non.

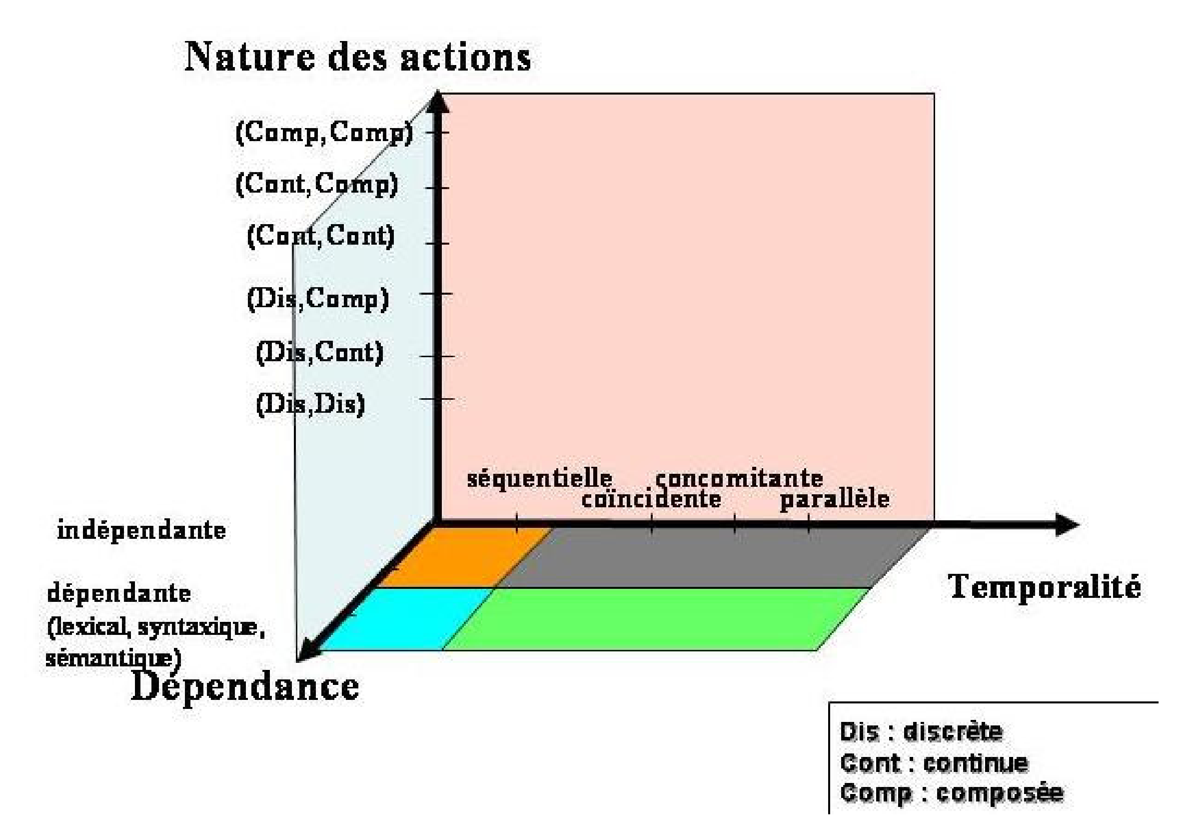


Figure 22: Espace de conception 2M pour l'interaction bi-manuelle

Tandis que les deux premières dimensions ont été étudiées dans le cadre de l'interaction bi-manuelle, la troisième dimension dépendance est à définir pour l'interaction bi-manuelle.

3. 3 Dimension Dépendance

Cette dimension considère la dépendance des sous-tâches effectuées par les deux mains. Comme la plupart des tâches sont composées et peuvent donc se décomposer en sous-tâches, il est difficile de déclarer la dépendance de sous-tâches sans fixer la granularité des sous-tâches considérées. En effet nous avons souligné au paragraphe 1, le continuum qui existe de l'action physique élémentaire à la tâche de plus haut niveau d'abstraction notée but à la Figure 16. Outre le problème de figer la granularité des sous-tâches considérées selon cette dimension, la notion même de dépendance est difficile à définir. Nous proposons ici deux définitions de la dépendance en adoptant deux points de vue distincts : l'un centré utilisateur, l'autre centré système.

3. 3. 1 Dépendance d’un point de vue utilisateur

Pour étudier la dépendance d'un point de vue utilisateur, nous considérons le résultat obtenu par la réalisation des deux sous-tâches. En effet une manière de définir la dépendance est d’évaluer le résultat des deux sous-tâches en fonction de l'usage, c'est-à-dire l’ordre dans lequel elles ont été effectuées. Soit A et B deux sous-tâches, nous considérons le résultat R selon l'ordre d'accomplissement de A et B :

- Soit R $_{A->B}$ le résultat de la sous-tâche A suivie de la sous-tâche B.
- Soit R $_{B->A}$ le résultat de la sous-tâche B suivie de la sous-tâche A.
- Soit R $_{A\&B}$ le résultat de la sous-tâche A effectuée en même temps que la sous-tâche B.

Tandis que A->B et B->A correspondent à un usage séquentiel selon la dimension temporelle de l'espace 2M, nous regroupons sous le terme A&B les cas d'entrelacement, de chevauchement et de parallélisme lors de la réalisation de A et B.

Nous définissons alors la dépendance par une relation d'équivalence :

Deux sous-tâches A et B sont dépendantes. ⇔ Les trois résultats R $_{A->B}$, R $_{B->A}$ et R $_{A\&B}$ ne sont pas identiques.

Nous déduisons de cette définition quatre cas. Le premier est l'indépendance. Les trois autres cas correspondent à trois formes de dépendance. Les quatre cas forment la complétude des relations de dépendance entre la sous-tâche A et la sous-tâche B :

1- Indépendance : R $_{A->B}$ = R $_{B->A}$ = R $_{A\&B}$

L'ordre des tâches n'a ici aucune importance. C'est le cas où l'utilisateur doit sélectionner deux cibles distinctes avec une souris dans chaque main. Les deux sous-tâches sont donc indépendantes.

2- Dépendance fusionnelle : R $_{A->B}$ = R $_{B->A}$ et R $_{A->B}$ ≠ R $_{A\&B}$

Faire la sous-tâche A puis la sous-tâche B ou le contraire ne modifie pas le résultat. Mais si les deux sous-tâches sont réalisées en même temps alors le résultat est différent. Ce cas correspond à l'exemple du pilotage d'un char dans un jeu de réalité virtuelle de la Figure 23: chaque main est associée à l'accélérateur d'une chenille. Si le joueur appuie sur l'accélérateur A puis sur l'accélérateur B, le résultat sera le même que si elle/il appuie d'abord sur le B puis sur le A. Comme l'illustre la Figure 23, dans les deux cas le char revient à sa position initiale. Cependant si le joueur appuie sur les deux accélérateurs en même temps, alors le char réagit différemment et avance en ligne droite.

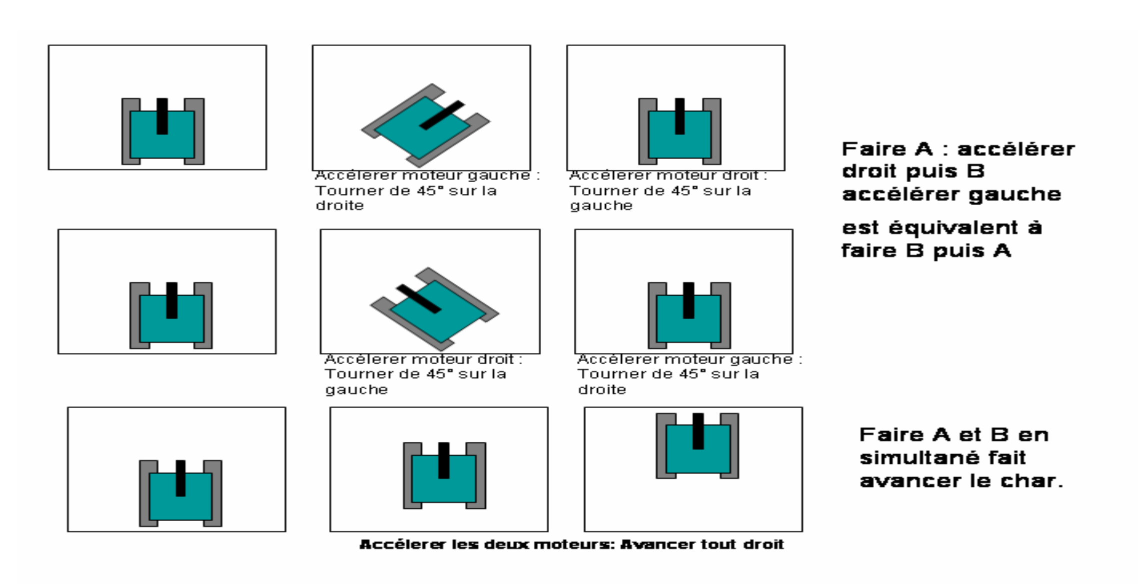


Figure 23 : Illustration du cas de dépendance fusionnelle : dépendance d'un char dans un jeu. A chaque main est associé l'accélérateur d'une chenille.

3- Dépendance temporelle : $R_{A\text{->}B} \neq R_{B\text{->}A}$ et $R_{A\text{->}B} = R_{A\&B}$
Le résultat $R_{A\&B}$ est équivalent à l'un des deux autres résultats. Mais ces derniers sont différents entre eux. Considérons un éditeur de dessin qui comprend une zone d'édition et une palette de couleur. Avec une main, l'utilisateur choisit la couleur, avec l'autre main, elle/il dessine avec la couleur courante. Dessiner avant ou après la sélection de la couleur va évidemment modifier la couleur courante du trait. Cependant faire les deux actions en même temps reviendra à faire l'un des deux cas précédents.

4- Dépendance temporelle et fusionnelle : $R_{A\text{->}B} \neq R_{B\text{->}A} \neq R_{A\&B}$
Les trois résultats sont distincts. Considérons l'éditeur de dessin bi-manuel [Yee 04] où l'utilisateur peut avec sa main non dominante déplacer la zone d'édition et de l'autre dessiner Figure 24). Comme l'illustre la Figure 24, les résultats sont bien différents.

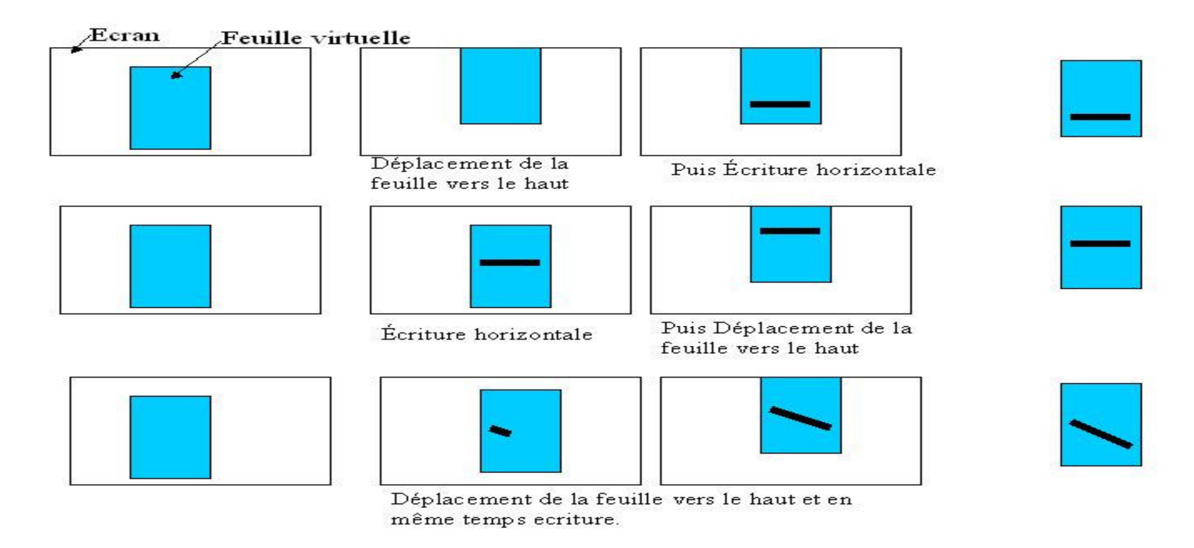


Figure 24 : Illustration du cas de dépendance temporelle et fusionnelle : éditeur de dessin bi-manuel. L'utilisateur fait glisser la zone d'édition avec sa main non dominante et dessine un trait avec sa main dominante.

3. 3. 2 Dépendance d'un point de vue système

Pour définir la dépendance de deux sous-tâches d'un point de vue système, nous reprenons la définition introduite dans le cadre de la multimodalité : la dépendance de deux sous-tâches implique la fusion, tandis que l'indépendance l'absence de fusion. Au paragraphe 2 du chapitre III, nous avons étudié la présence ou l'absence de fusion multimodale en considérant les aspects temporels pour obtenir quatre classes d'usage de modalités : Exclusif, Concurrent, Alterné et Synergique.

En considérant les niveaux d'abstraction de la Figure 15, nous définissons alors trois niveaux de fusion : lexicale, syntaxique et sémantique. La fusion lexicale s'opère entre les niveaux dispositif et langage, celle syntaxique entre les niveaux langage et tâche et celle sémantique entre les niveaux tâche et but.

- **La fusion lexicale** traduit une synchronisation entre des informations de bas niveau. Elle correspond à l'aspect articulatoire de l'espace de la Figure 15. Par exemple, l'enfoncement conjoint de la touche “shift” et d'un clic souris est

fusionné par le système en un seul événement. En java, par exemple l'événement MouseEvent possède un champ qui indique la position, mais aussi les touches enfoncées du clavier. Cette fusion correspond à l'opérateur de composition de l'espace des dispositifs de Mackinlay (paragraphe 1 du chapitre III).

- **La fusion syntaxique** consiste à combiner des informations pour obtenir une commande complète ou tâche élémentaire, comme le paradigme du "mets ça là" [Bolt 82].
- **La fusion sémantique** combine des commandes ou tâches élémentaires pour aboutir à un nouveau résultat ou but. Par exemple, dans VoicePaint [Gourdol 92], un éditeur de dessin qui permet de dessiner à la souris tout en modifiant à la voix les caractéristiques du pinceau (peinture, épaisseur, luminosité), deux commandes (dessiner une droite et changer de couleur) sont combinées en une seule pour obtenir une droite à deux couleurs. Nous trouvons dans l'éditeur de dessin bi-manuel [Yee 04] présenté à la Figure 28 un cas très similaire : la parole exploitée dans VoicePaint est remplacée par une sélection de la main non-dominante.

3. 3. 3 Conclusion sur la dimension dépendance

Nous avons introduit deux définitions de la dépendance des sous-tâches effectuées par les deux mains, l'une centrée utilisateur, l'autre système. Dans notre espace 2M, nous adoptons la définition système, qui est plus générale, en considérant l'absence de fusion ou la présence de fusion lexicale, syntaxique et sémantique (Figure 15). En effet, les dépendances définies d'un point de vue utilisateur peuvent être réalisées par une fusion lexicale, syntaxique et/ou sémantique. Tandis que le point de vue utilisateur se base sur le résultat obtenu c'est-à-dire le niveau "but" de la Figure 15, le point de vue système étudie les lieux de fusion dans le mécanisme d'abstraction qui des actions utilisateurs déduit un but ou état système. Il n'existe donc pas une relation bijective entre les dépendances définies d'un point de vue utilisateur et les trois types de fusion identifiés.

Les axes de l'espace 2M exposés, nous analysons maintenant ses apports en appréciant son pouvoir taxinomique mais aussi ses capacités génératives et comparatives utiles en phase de conception.

3. 4 Apports de l'espace 2M

Au paragraphe 3.1, nous avons rappelé les trois apports possibles d'un espace de conception. Dans ce paragraphe, nous les considérons successivement en commençant par étudier le pouvoir taxinomique de l'espace 2M puis en considérant ses apports en phase de conception c'est-à-dire son pouvoir génératif et comparatif.

3. 5 Pouvoir taxinomique

La classification des techniques d'interaction bi-manuelle existantes permet de regrouper les systèmes bi-manuels aux caractéristiques communes. Il permettra également de situer les futurs systèmes bi-manuels par rapport à l'existant. Pour illustrer le pouvoir taxinomique de notre espace, nous considérons plusieurs techniques d'interaction bi-manuelle que nous situons dans notre espace 2M.

3. 5. 1 Défilement et sélection

Pour la sélection d'une citation (ou lien hypertexte) dans un document, deux sous-tâches ont été très souvent étudiées dans le cadre de l'interaction bi-manuelle : le défilement du document et la sélection (pointer-cliquer). Par exemple, dans [Zhai 97], il a été montré qu'il n'y a pas d'avantage significatif pour l'interaction bi-manuelle de la Figure 25 qui comprenait une souris dans la MD pour pointer-cliquer et un joystick (position isotonique) dans la MND pour défiler les pages. La comparaison avec l'interaction à une main était réalisée avec trois dispositifs : une souris, une souris à mollette et la souris d'IBM avec un joystick.



Figure 25 : Système bi-manuel avec un joystick dans la main non dominante et une souris dans la main dominante pur la tâche de défilement-sélection [Zhai 97]

Dans [Myers 00] l'exploitation d'un ordinateur de poche (PDA) comme deuxième dispositif d'interaction (Figure 26) est étudiée, en partant du constat que ce dernier est souvent placé à côté de l'ordinateur et lui est connecté. Ce dispositif est plus général que beaucoup de dispositif grâce à son écran tactile. Un désavantage lié à cela est qu'il n'est pas forcément naturel pour l'utilisateur car les actions portent sur des objets affichés comme des boutons logiques et non sur la manipulation de boutons physiques. Le temps pour passer du clavier au PDA + souris est à peine supérieur de 10% au temps pour passer du clavier à lasouris.

Le PDA était manipulé par la MND pour le défilement, tandis que la souris dans la MD pour la sélection. Plusieurs techniques de défilement ont été conçues sur le PDA : des boutons, une barre de défilement absolu et une technique se comportant comme un joystick à trois vitesses. Pour cette dernière technique, l'utilisateur en glissant le doigt fait partir le défilement dans la direction indiquée, l'amplitude du mouvement indiquant la vitesse. Les résultats sont très proches voir meilleurs que ceux reportés dans [Zhai 97].



Figure 26: Système bi-manuel avec un PDA dans la main non dominante et une souris dans la main dominante pour la tâche de défilement-sélection [Myers 00]

Plus récemment dans [Mcloone 03] le clavier est augmenté de touches spéciales sur le côté gauche pour favoriser l'interaction bi-manuelle. L'une d'entre elles est une roue pour le défilement, les autres sont des raccourcis pour Copier/Couper/Coller, pour naviguer entre les différentes applications (CTRL + TAB). Le changement de clavier n'augmente pas le coût cognitif car très naturel. Les utilisateurs ont beaucoup apprécié ce nouveau clavier.



Figure 27 : Clavier microsoft augmenté avec une roue pour faire défiler [MCLONNE 03]

L'articulation des deux sous-tâches défiler par la MND et sélectionner par la MD respecte la propriété énoncée dans [Buxton 05] : la main non dominante fixe le contexte, tandis que la manipulation proprement dite est effectuée par la main dominante. Au sein de notre espace, cette interaction bi-manuelle se caractérise par :

Temporalité : les sous-tâches sont effectuées de façon séquentielle, et parfois concomitante pour un utilisateur expert.

Nature des actions : (MND-continue sauf pour les boutons sur le PDA [MYERS 00] où l'action était discrète MND-dicrète, MD-composée).

Dépendance : les sous-tâches correspondent à des commandes distinctes qui n'impliquent pas de fusion. Les sous-tâches sont donc indépendantes bien qu'elles concourent à un but commun.

3. 5. 2 Editeur de dessin [Yee 04]

Dans [Yee 04], une surface bi-manuelle est créée en superposant un écran tactile au-dessus d'une tablette-PC comme le montre la Figure 28. Deux tâches de dessin sont considérées pour l'interaction bi-manuelle.

Une première tâche de dessin est décomposée en deux sous-tâches, l'une de déplacement de la feuille de dessin par la MND, l'autre de dessin d'un trait par la MD. Cette tâche se rapproche de celle considérée précédemment, la MND fixant le contexte dans lequel la manipulation est faite par la MD [Buxton 05]. Cette interaction bi-manuelle a donc la même position dans notre espace 2M que le cas précédent.



Figure 28 : Tâche de dessin [YEE 04]. Il est possible de sélectionner la couleur ou déplacer la feuille virtuelle avec la main non dominante et dessiner avec la main dominante.

Une autre tâche de dessin est décomposée en deux sous-tâches, l'une de sélection de la couleur dans la palette par la MND, l'autre de dessin d'un trait par la MD. Cette interaction bi-manuelle permet d'obtenir un trait multicolore. Au sein de notre espace, elle est caractérisée par:

Temporalité : les sous-tâches sont effectuées de façon séquentielle mais aussi coïncidente, la sous-tâche de changements de couleurs par la MND étant effectuée pendant la réalisation de la sous-tâche de dessin du trait.

Nature des actions : (MND-discrète, MD-composée).

Dépendance : les sous-tâches correspondent à des commandes distinctes qui sont fusionnées au niveau sémantique pour obtenir un trait multicolore. Les deux sous-tâches sont donc dépendantes et leur fusion est effectuée au niveau sémantique. Ce cas correspond à l'interaction dans VoicePaint [Gourdol 91] où la couleur du trait était changée par la voix.

3. 5. 3 Toolglass

Une toolglass [Bier 94] est une surface transparente, sur laquelle sont disposés des outils (interacteurs) semi-transparents, comme le montre la Figure 29. La fenêtre de la toolglass est manipulée de la MND : elle est "attachée" à une trackball (ou tout autre périphérique échantillonnant des trajectoires planes) qui permet de déplacer la toolglass dans les deux dimensions du plan, afin de pouvoir superposer les outils qu'elle offre avec des objets de l'application situés en dessous. Ce système définit trois couches: le pointeur, manipulé avec la MD, la toolglass, portant les outils et contrôlée avec la MND et les objets de l'application en dessous. Pour utiliser un des interacteurs de la toolglass, l'utilisateur superpose l'outil à utiliser, l'objet à modifier et le pointeur.

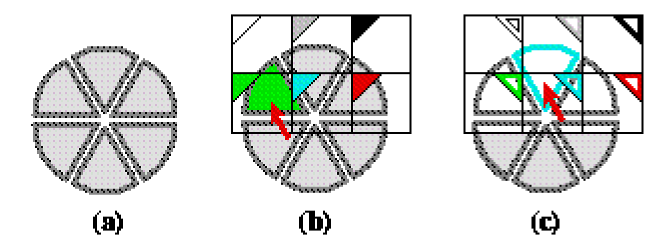


Figure 29 : Toolglass [Bier 04]

Cette interaction bi-manuelle revient au premier cas étudié : la main non dominante fixe le contexte, tandis que la manipulation proprement dite est effectuée par la main dominante [Buxton 05]. Cependant au niveau de notre espace 2M, nous distinguons les deux cas au niveau de la dimension Dépendance :

Temporalité : les sous-tâches sont effectuées de façon séquentielle, et parfois concomitante pour un utilisateur expert.

Nature des actions : (MND-continue, MD-composée).

Dépendance : les sous-tâches sont dépendantes et impliquent une fusion syntaxique. En effet tandis que le déplacement de la toolglass spécifie les paramètres de la commande (par exemple couleur et objet), celle-ci est déclenchée par la MD. Ce cas s'apparente à celui de l'interaction multimodale où l'utilisateur désigne une icône et énonce la commande vocale "détruire".

Il convient de noter que l'interaction bi-manuelle avec la toolglass et celle de l'éditeur de dessin du paragraphe précédent pour effectuer une même tâche comme le coloriage sont différentes car les sous-tâches affectées aux deux mains ne sont pas les mêmes. Dans le cas de l'éditeur de dessin [Yee 04], la MND sélectionne la couleur tandis que la MD désigne l'objet à colorier. Dans le cas de la toolglass, la MND déplace la palette de couleurs et spécifie ainsi la couleur et l'objet tandis que la MD déclenche la tâche.

3. 5. 4 Utilisation de touches spéciales

Dans les interfaces graphiques classiques, les utilisateurs experts ont naturellement évolué vers l'interaction bi-manuelle en utilisant en même temps que la souris dans la MD, les raccourcis clavier (par exemple, Ctrl-C, Ctrl-X, Ctr-V) dans la MND. Dans [Mcloone 03], le clavier comprend des touches dédiées pour un ensemble de raccourcis comme le montre la Figure 27. Notons q'un clavier augmenté des principaux raccourcis avait été proposé dès les années 1980.

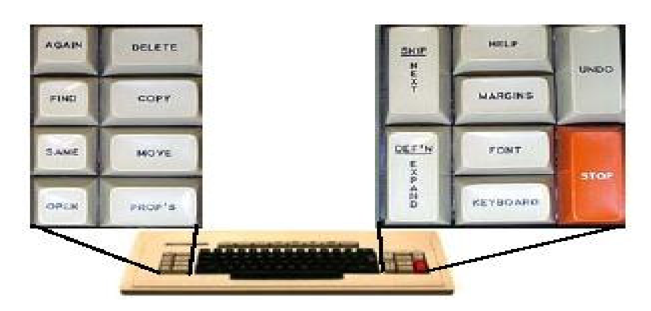


Figure 30 : Clavier Xerox avec des raccourcis sur la gauche pour la main non dominante

De même dans [Myers 00], le PDA est utilisé pour afficher des boutons de raccourcis, sélectionnés par la MND. La SNCF utilise aussi un petit boîtier avec 4*4 touches pour avoir un grand nombre de fonctionnalités accessibles rapidement. Ce boîtier oblige un apprentissage pour les utilisateurs.

La MND effectue une action discrète. En ne faisant aucune hypothèse sur le type d'action effectuée par la MD (discrète, continue ou composée), au sein de l'espace 2M, la caractérisation diffère selon les dimensions Dépendance et Temporalité en considérant deux rôles de la touche. (1) Si la touche a un rôle de modifieur comme Shift dans la sélection multiple, les deux sous-tâches sont dépendantes et leur fusion est du niveau lexical. De plus les sous-tâches sont effectuées de façon coïncidente. (2) Si la touche a un rôle de raccourci d'une sous-tâche fréquente, les valeurs selon les dimensions Dépendance et Temporalité dépendent de la tâche considérée. Par exemple selon la dimension Dépendance, les sous-tâches peuvent être (indépendantes, dépendantes avec fusion syntaxique, dépendantes avec fusion sémantique).

3. 5. 5 Interfaces palpables

Au sein des interfaces palpables (Graspable) [Fitzmaurice 95] des objets physiques saisissables servent de dispositifs d'entrée. La position et l'orientation de ces objets sont à tout instant connues du système.

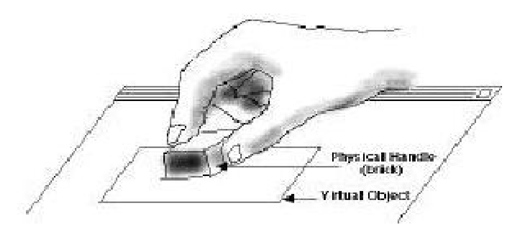


Figure 31 : Une interface palpable avec des briques saisissables [Fitzmaurice 95]

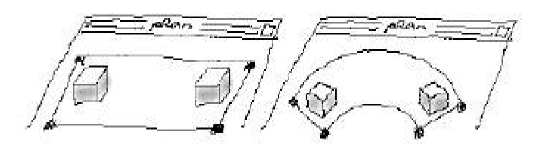


Figure 32 : Exemple de manipulation des briques (translation, rotation) dans une interface palpable [Fitzmaurice 95]

Comme la manipulation d'objets est plus directe et plus maniable, les interfaces palpables encouragent l'interaction bi-manuelle, la simultanéité des tâches et la collaboration. Le prototype Briques par exemple, utilise des briques « lego » de jouets d'enfant pour dessiner des formes géométriques simples. L'interaction bi-manuelle est exploitée lors la création, la déformation et le placement de formes géométriques.

Au sein d'une interface classique, pour réaliser une commande, il y a trois phases :

Acquérir le dispositif physique (ex : une souris), acquérir le dispositif logique (ex : prendre la barre de défilement avec la souris) et activer la fonction logique (ex : déplacer la barre de défilement afin de faire défiler des pages Web). Grâce aux interfaces palpables, la seconde étape (acquérir le dispositif logique) est supprimée car elle correspond au dispositif physique.

Dans [Hinckley 94] une interface qui est bi-manuelle est proposée pour la visualisation en neurochirurgie, comme le montre la Figure 33. L interaction à deux mains aide à la manipulation d'images médicales (un scanner IRM de cerveau). Dans la main non dominante, le chirurgien tient une tête miniature (fixation du contexte) et dans l'autre main un outil de sélection qui peut glisser sur la miniature.



Figure 33 : Une interface palpable de visualisation pour la neurochirurgie [Hinckley 94]

Au sein de l'espace 2M, cette interaction bi-manuelle se caractérise comme la plupart des interfaces palpables par :
Temporalité : les sous-tâches sont souvent effectuées de façon parallèle.
Nature des actions : (MND-continue, MD-continue).
Dépendance : Cas de fusion syntaxique où les deux objets fournissent l'orientation du vecteur d'observation.

En considérant plusieurs techniques d'interaction bi-manuelles existantes, outre l'illustration de l'espace 2M, nous avons démontré son pouvoir taxinomique.

3. 6 Pouvoir génératif et comparatif

Tandis que le positionnement de nombreux systèmes existants au sein de l'espace 2M a permis de souligner son pouvoir taxinomique, montrer ses apports en phase de conception (pouvoir comparatif d'alternatives de conception et génératif de nouvelles formes d'interaction) est beaucoup plus difficile. Cependant l'espace 2M par l'identification de trois axes (et des valeurs discrètes sur chaque axe) permet de guider la conception par l'exploration systématique de l'espace des solutions. Les valeurs sur les axes définissent des choix à effectuer en phase de conception. Notons que la complétude de l'espace n'est pas prouvée : des choix de conception peuvent ne pas être pris en compte par l'espace et d'autres axes sont peut-être nécessaires.

Pouvoir comparatif

Néanmoins l'espace ne permet pas de guider le concepteur pour choisir entre plusieurs alternatives de conception localisées en des points différents de l'espace 2M. Pour augmenter son pouvoir comparatif, l'espace 2M doit donc être enrichi par une mesure de la qualité ergonomique de chaque point de l'espace. Ceci peut se faire dans un premier temps en considérant des tâches bi-manuelles et nécessite le développement et l'évaluation expérimentale de techniques d'interaction bi-manuelle. Il conviendra alors de conclure sur la généralité des résultats indépendamment des tâches considérées.

Pouvoir génératif de nouvelles formes d'interaction

L'identification d'un ensemble de paramètres de conception au sein de l'espace incite le concepteur à considérer des alternatives en changeant un ou plusieurs paramètres à une technique conçue. En cela, l'espace 2M peut favoriser la conception de nouvelles formes d'interaction. Nous illustrerons ce point dans la partie II où nous appliquons notre espace 2M à la conception d'un système bi-manuel. Néanmoins il est toujours difficile de mesurer l'impact d'un espace de conception sur la solution conçue. Il conviendrait d'appliquer notre espace à de nombreux systèmes et de solliciter des équipes de concepteurs différentes. Par exemple un exercice intéressant à mener réside dans une étude comparative, en observant deux équipes de concepteurs sur un même projet, l'un utilisant l'espace 2M et l'autre non, puis en comparant les solutions de conception obtenues.

4 Conclusion sur l'espace 2M

En conclusion, la capitalisation de plusieurs résultats en interaction multimodale et bi-manuelle des chapitres II et III est à l'origine de notre nouvel espace de conception 2M dédié à l'interaction bi-manuelle comme le schématise la Figure 34.

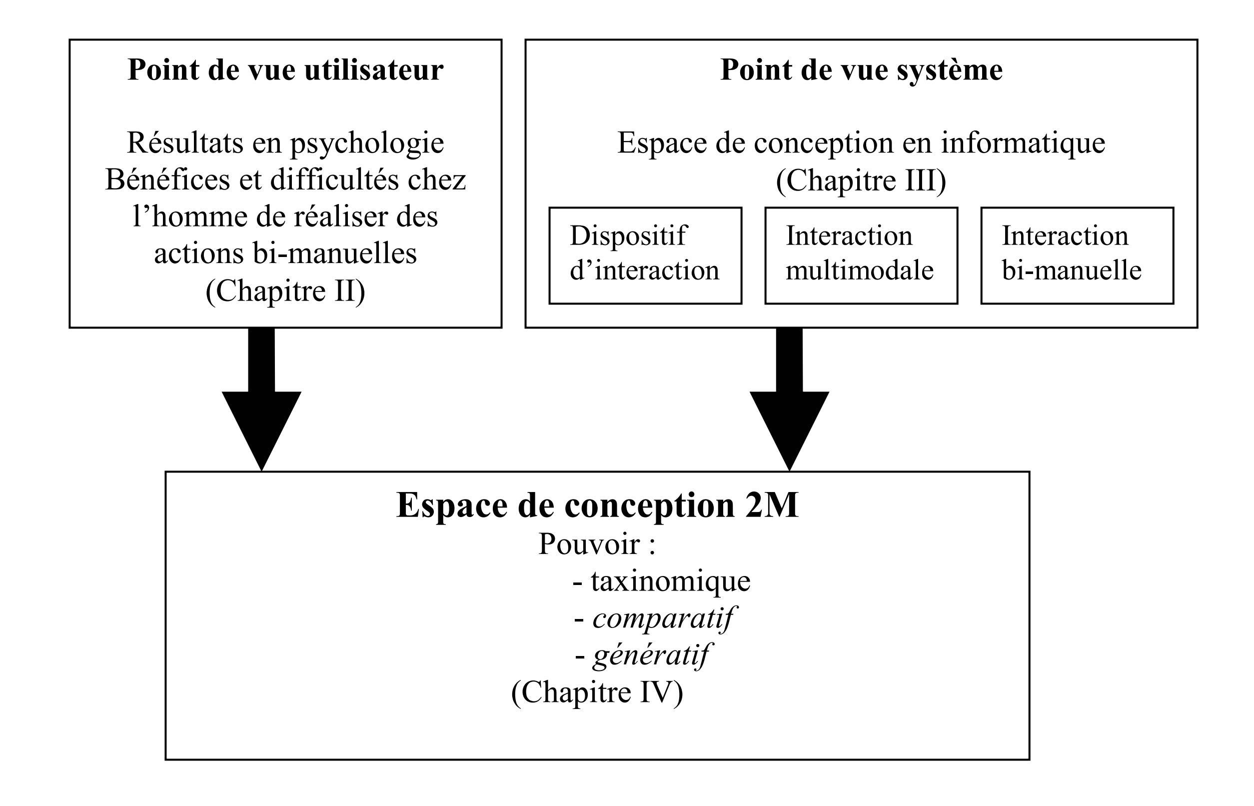


Figure 34 : Synthèse de la démarche de travail pour l'espace 2M

L'espace 2M s'articule selon trois axes pertinents pour la conception de l'interaction bi-manuelle qui sont :

- le type d'actions effectuées par chaque main,

- les relations temporelles entre les actions des deux mains,
- la dépendance des actions de chaque main.

En particulier, nous avons mené une étude détaillée des dépendances des actions de chaque main tant du point de vue utilisateur que système. Le choix d'adopter le point de vue système dans notre espace 2M se justifie par sa généralité, les différents types de fusion identifiés pouvant exprimer les dépendances du point de vue utilisateur. Néanmoins la mise en relation de ces deux points de vue offre des perspectives de recherche à approfondir : pour cela il convient d'étudier les deux flots de données issus des actions des deux mains au regard des niveaux d'abstraction et des lieux de fusion de la fonction d'abstraction des actions physiques en un but utilisateur.

Comme pour tout espace de conception, la pertinence de l'espace 2M doit être étudiée selon son pouvoir taxinomique mais aussi son apport lors de la phase de conception de l'interaction en mesurant son pouvoir comparatif d'alternatives de conception et génératif de nouvelles formes d'interaction. Dans ce chapitre, nous avons prouvé l'apport de l'espace 2M en classant plus de vingt systèmes bi-manuels. L'intérêt de l'espace 2M en conception est plus difficile à prouver car il conviendrait de l'utiliser pour différents projets et de faire intervenir plusieurs équipes de concepteurs. Néanmoins dans le cadre de notre étude, un premier pas pour mesurer le pouvoir comparatif et génératif de l'espace 2M est de l'appliquer à la conception d'un système bi-manuel. Ceci fait l'objet de la partie II de mon rapport où nous considérons la conception et la réalisation d'un système bi-manuel pour manipuler des graphes d'ARN.

Partie II

Application : Manipulation bi-manuelle de graphes biologiques

Chapitre V Application de nos résultats à la manipulation bi-manuelle de graphes biologiques

Nous illustrons l'application de notre espace 2M à la conception d'un système bi-manuel. L'un de nos objectifs est d'apprécier le pouvoir comparatif et génératif de l'espace 2M en phase de conception. Le domaine d'application est la biologie et nous focalisons sur le travail d'un groupe de biologistes en train d'étudier des graphes d'ARN.

Tandis que ce chapitre explique l'analyse de l'activité que nous avons menée auprès de biologistes et liste les tâches informatiques que nous avons déduites de cette analyse, le chapitre suivant présente la conception de techniques d'interaction bi-manuelle pour ces tâches et leur réalisation logicielle.

Dans ce chapitre, nous présentons les résultats de la collaboration avec les biologistes. Nous décrivons d'abord les structures d'ARN manipulées afin de nous familiariser avec le contexte biologique. Nous exposons ensuite la démarche de collaboration avec les biologistes à travers une analyse ascendante et descendante qui nous a permis d'identifier leurs besoins et les tâches informatiques qui en résultent.

1 Structures biologiques manipulées

Ce paragraphe a été repris et complété du rapport de Nicolas Gandin [Gandin 04].

Les structures manipulées concernent l'Acide Ribonucléique, ARN. L'acide Ribonucléique est une molécule très importante qui accomplit un grand éventail de fonctions dans le système biologique. On le retrouve dans le noyau, où il est synthétisé, et dans le cytoplasme, comme :

- ARN messager ou ARNm (qui transporte l'information en dehors du noyau),
- ARN de transfert ou ARNt (qui décode l'information),
- ARN ribosomal ou ARNr (qui constitue avant le ribosome, un assemblage de molécules).

Ces différentes formes d'ARN sont invoquées dans la synthèse de protéines et d'autres fonctions de régulation..

Par exemple, c'est l'ARN qui contient l'information génétique du virus comme l'HIV et de plus régule les fonctions de celui-ci. L'ARN est récemment devenu le centre d'attention des biologistes, de par ses propriétés catalytiques, qui se traduit par le besoin d'obtenir des informations structurelles. Structurellement un ARN est un polymère dans lequel des monomères (ribonucléiques) sont liés ensemble par des ponts. Chaque nucléotide contient une base. Les quatre bases sont :

- Ademine (A)
- Cytosine (C)
- Guanine (G)
- Uracil (U)

Un ARN est constitué d'une seule et même chaîne polynucléique, qui est loin d'être régulièrement linéaire. Il comporte de vastes régions de bases appariées. Les bases

complémentaires A-U et G-C forment des appariements de bases stables à travers la création de liens d'hydrogène entre les sites récepteurs et émetteurs de ces bases. Ils sont appelés appariements de type Watson-Crick. D'autres appariements existent, G-U (la paire branlante) comme les deux précédentes est une paire canonique, contrairement aux appariements non canoniques comme (A-U et G-C). Il existe trois façons de représenter la structure d'ARN :

- La structure primaire de l'ARN qui est la représentation de l'ARN comme une simple séquence de nucléotides.
- La structure secondaire (visualisation en 2 dimensions) d'ARN réfère à des motifs structuraux comme les tiges, boucles en épingle à cheveux, boucles intérieures et multi-branches. Les tiges sont des régions auto-complémentaires (A-U, U-A, G-C, C-G), tandis que les boucles en épingle à cheveux ont des régions avec des bases non appariées. Les jonctions d'ARN (boucles intérieures et multi-branches) se créent quand deux ou plusieurs tiges se rencontrent, et qu'elles contiennent des bases qui ne correspondent pas. L'architecture globale de la molécule est principalement stabilisée par les paires de bases canoniques A-U, G-C et G-U. Les derniers résultats en biologie montrent que cette représentation atteint ses limites et qu'une grande quantité d'information est perdue lorsque que l'on passe de la représentation 3D (structure tertiaire) à la structure secondaire
- La structure tertiaire (structure en 3D) est de plus en plus utilisée car il semblerait que ce ne soit pas la séquence de nucléotide qui soit le plus important, mais la structure de l'ARN dans l'espace. La structure de l'ARN définit sa fonctionnalité. Si deux ARNs n'ont pas la même séquence, mais qu'ils ont la même structure dans l'espace, il est très probable que leurs fonctionnalités soient proches.

2 Démarche de collaboration avec les biologistes

Il est évident que la biologie n'est pas notre domaine de compétence. Il était donc indispensable pour nous de collaborer afin de pouvoir comprendre leur travail et leur contexte de travail (analyse de l'activité). Cette première approche centrée utilisateur est appelée analyse descendante. A l'inverse, l'analyse ascendante consiste à proposer aux biologistes de nouveaux outils déjà élaborés ou en cours d'élaboration afin de valider les techniques d'interaction et de créer chez eux de nouveaux besoins. Ces deux types d'analyse sont complémentaires.

2. 1 Analyse descendante

Nous voulions très tôt se rapprocher des biologistes afin de bien comprendre leurs besoins et d'élaborer un système qui soit en adéquation avec leur travail. Pour cela, il a fallu mener une analyse des besoins et une analyse de l'activité. Ceci s'est fait dans le cadre du projet ACI NavGraphe qui regroupe des biologistes, des bio-informaticiens et des informaticiens en IHM. Les biologistes du projet nous ont proposé d'assister à la réunion ACI ARENA, projet regroupant des biologistes et des bio-informaticiens. Beaucoup de biologistes étaient présents ce qui nous a permis de comprendre leurs besoins pour l'analyse des structures secondaires d'ARN.

2. 2 Analyse ascendante

Il est important que la collaboration se fasse dans les deux sens. En effet, les premières discussions ont montré qu'il était difficile pour eux de concevoir ce que nous pouvions faire et ce qu'étaient nos motivations. Il était donc indispensable que l'on puisse leur expliquer clairement nos sujets de recherche et comment nous pouvions apporter nos compétences à l'étude et à la manipulation de l'ARN.

Pour cela, lors de l'ACI NavGraphe et de l'ACI ARENA, nous avons présenté le contexte de travail que nous voulions proposer, la table augmentée et le multi-surface. L'objectif affiché est de pouvoir visualiser une grande quantité d'informations car les graphes biologiques sont de très grandes tailles, d'autoriser la manipulation d'un graphe projeté sur la table avec des jetons et de favoriser la collaboration (être à plusieurs autour de la table). Comme nous n'avions pas développé les techniques de manipulation, nous avons décidé d'élaborer des story-boards qui leur permettraient de se rendre compte sur des cas concrets les nouvelles possibilités qui s'offraient à eux. Nous avons donc fait appel à un dessinateur Romain Boussard. Il y a eu un véritable travail de collaboration. En effet, il fallait d'une part lui faire comprendre ce que nous voulions qu'il dessine. D'autre part, nous retravaillions ses dessins, en insérant dans ces derniers de véritables graphes biologiques à l'aide de PhotoShop. Ces story boards ont été très utiles, car ils ont permis de présenter des techniques d'interaction qu'il n'était pas toujours facile d'expliquer. Ces story-boards sont présentés en annexe de ce rapport.

3 Tâches identifiées

Les résultats de la collaboration avec les biologistes bien que difficile au départ ont été globalement fructueux et nous ont permis d'identifier quatre tâches où un utilisateur peut manipuler avec des jetons un objet projeté sur la table.

3. 1 Tâche d'édition

Certains biologistes semblaient intéressés par le fait de pouvoir éditer la structure secondaire d'ARN surtout à des fins pédagogiques. Il semble qu'ils ne disposent pas actuellement de bons logiciels d'édition. Ils souhaitent pouvoir :

- Supprimer, rajouter ou supprimer une base.
- Modifier les liaisons entre les bases pour créer des boucles.
- Mettre en évidence certaines parties de la molécule d'ARN.
- Annoter des informations.
- Déformer certaines parties du graphe afin de se rapprocher d'une autre structure.

Toutes ces opérations d'édition doivent respecter des critères d'esthétisme fixés par les biologistes. Ces critères sont :

- Tailles des arêtes constantes.
- Représentation circulaire des bases non appariés.
- Pas de recouvrement (les arêtes ne doivent pas se couper).

Il s'agit donc de proposer des techniques d'édition qui respectent ces critères d'esthétisme.

3. 2 Tâche de comparaison

Les biologistes ont souvent besoin de comparer les structures d'ARN venant de la même famille. Pour cela, ils aimeraient pouvoir les superposer et observer une mise en évidence des différences entre les graphes d'ARN. Il faut pour cela que deux graphes d'ARN qui ont une structure proche puissent avoir une représentation graphique proche. En effet d'un point de vue biologique, deux espèces qui ont une structure d'ARN proche ont des fonctionnalités communes.

3. 3 Tâche d'évolution

Les biologistes aimeraient bien pouvoir à partir d'une structure A, d'une structure B et d'un certain langage d'interaction pouvoir visualiser quelle pourrait être la transformation qui a amené la molécule A à se transformer en la molécule B, cela surtout dans un but pédagogique. En terme informatique, il s'agit de pouvoir à partir d'un graphe A initial et du même graphe édité par les biologistes, de reconstruire de manière interactive les différentes étapes qu'ont réalisées les biologistes pour aboutir au graphe final, en quelque sorte un « morphing » avec contrainte sur deux structures.

3. 4 Tâche de visualisation multiple

Nous avons appris qu'ils allaient progressivement abandonner la structure secondaire pour passer à la visualisation en 3D qui apporte beaucoup d'informations sur la structure de l'ARN et donc sur ses fonctionnalités. Cependant, en continuant de discuter, les biologistes aimeraient un système capable de maintenir une représentation de la structure secondaire et de la structure tertiaire en même temps. Ils aimeraient pouvoir sélectionner une partie de la molécule d'ARN sur la représentation secondaire et voir cette sélection en 3D sur un autre écran.

4 Conclusion sur la collaboration avec les biologistes

En conclusion de ce bref chapitre consacré à notre collaboration avec les utilisateurs finaux de notre système bi-manuel, les biologistes, nous soulignons l'approche à la fois descendante et ascendante adoptée pour identifier leurs besoins et tâches. Ainsi nous avons combiné interviews et discussions pour comprendre leur travail avec l'exposé de techniques d'interaction que nous avons concrétisées sous la forme de storyboards. De cette collaboration, nous retenons un ensemble de tâches d'édition, de comparaison d'évolution et de visualisation multiple. L'étape suivante consiste à concevoir des techniques d'interaction adaptées à ces tâches, puis à les développer. Ceci fait l'objet du chapitre suivant.

Chapitre VI Techniques d’interaction conçues et développées

Dans ce chapitre, nous exposons la conception et la réalisation de techniques d'interaction bi-manuelle pour les tâches présentées au chapitre précédent. Ce chapitre se décompose en trois parties respectivement dédiées à :

- la conception de l'interaction où nous soulignons l'apport de l'espace 2M,
- la conception logicielle selon le modèle d'architecture logicielle PAC-Amodeus,
- la phase de codage où nous mettons en avant la qualité du code (mécanisme systématique de codage, documentation, extensibilité).

1 Description de l’application

Le système NavGraphe permet à des biologistes d’explorer, de manipuler, d’éditer et de comparer des structures d’ARN. Nous avons réalisé des «storyboards » joints en annexe afin d’illustrer un scénario d'usage. Pour favoriser la collaboration autour d'un graphe d'ARN, pour ne pas limiter l'interaction à un seul utilisateur et pour encourager l'interaction bi-manuelle, nous avons choisi de projeter le graphe sur une table et de le manipuler avec des jetons reconnus par une caméra [Bérard 99]. Outre l'interaction en entrée, la grande taille des graphes manipulés justifie le choix d'une grande surface d'affichage comme la table. La Figure 47 présente le contexte d'usage du système NavGraphe. Notons à la Figure 47 le fait que d'autres surfaces peuvent être intégrées pour étendre la table, comme les murs ou un ordinateur de poche. Nous entrevoyons la conception d'un véritable environnement augmenté informationnel.

Le contexte d'usage fixé, nous expliquons maintenant la conception des techniques d'interaction. Au paragraphe suivant, par décomposition hiérarchique des tâches de haut niveau du chapitre précédent, nous identifions les tâches élémentaires d'interaction. Nous décrivons ensuite les techniques d'interaction conçue pour réaliser ces tâches élémentaires.

1. 1 Arbre des tâches

La décomposition hiérarchique des tâches de NavGraphe en tâches élémentaires est décrite avec la notation HTA dont les opérateurs principaux sont rappelés à la Figure 34.

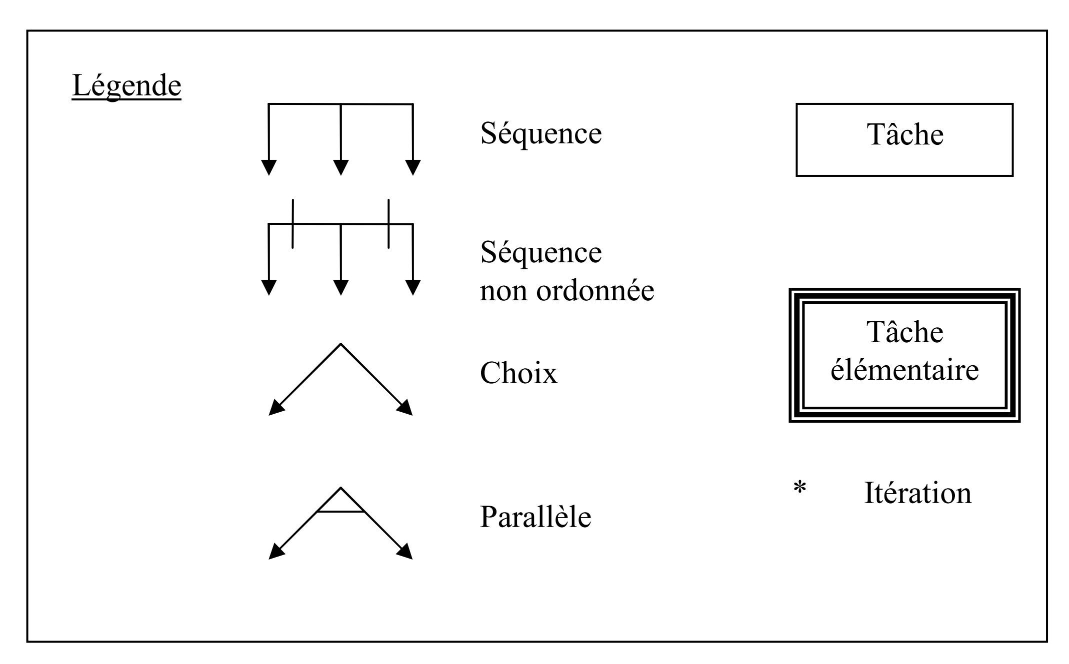


Figure 34 : La notation HTA

L'application NavGraphe se décompose en deux tâches séquentielles :

- Utilisation de l'application
- Quitter l'application

La Figure 35 montre ces deux tâches séquentielles qui sont affinées dans les paragraphes suivants.

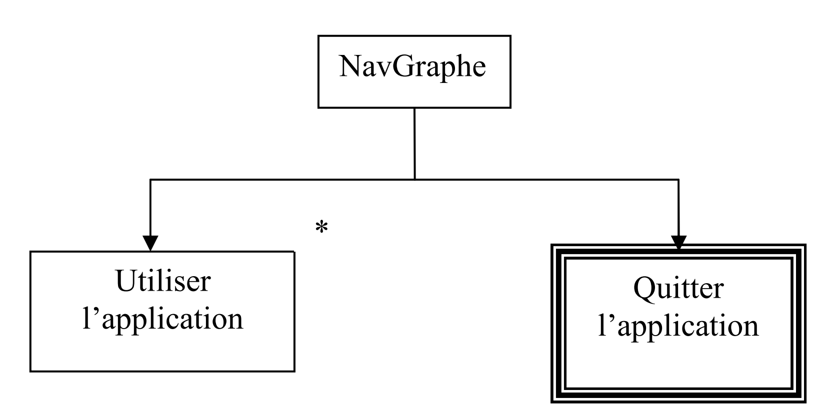


Figure 35 : Système NavGraphe

Comme le montre la Figure 36, l'utilisation de l'application se décompose en deux tâches : Etudier un graphe ou étudier un sous-graphe.

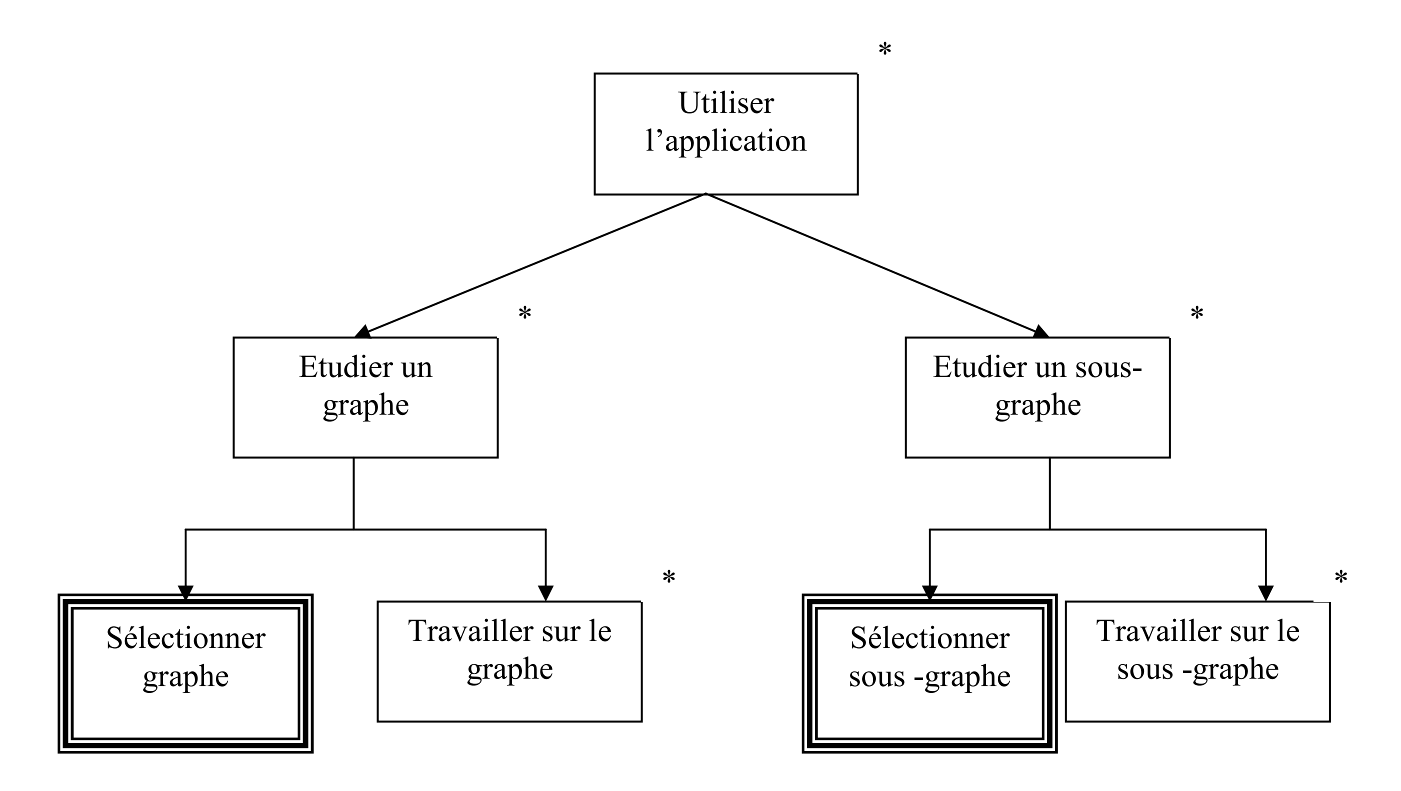


Figure 36 : Tâche utiliser l'application

La tâche « Travailler sur le graphe » décrite à la Figure 37 est décomposée en plusieurs tâches :

- Manipuler le graphe
- Editer le graphe et,
- Déformer le graphe.

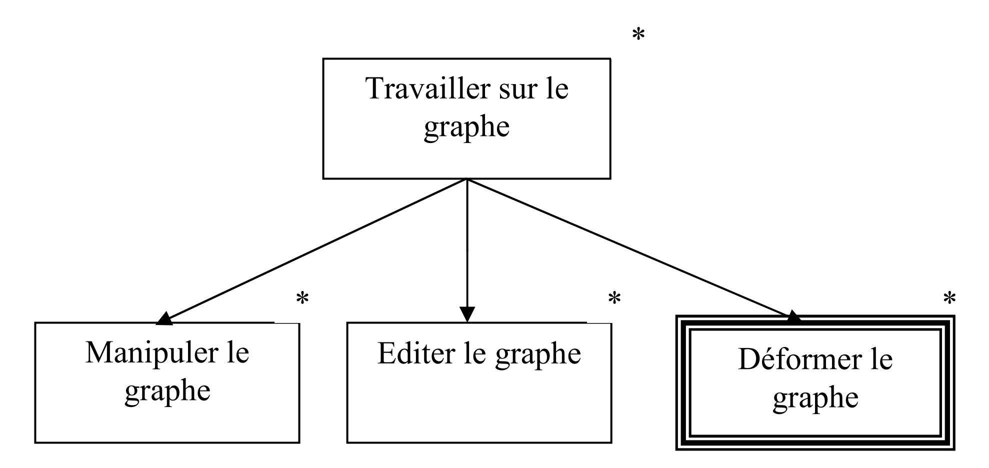


Figure 37 : Tâche travailler sur le graphe

Manipuler le graphe consiste à pouvoir lui faire subir une translation, une rotation ou un zoom comme le montre la Figure 38.

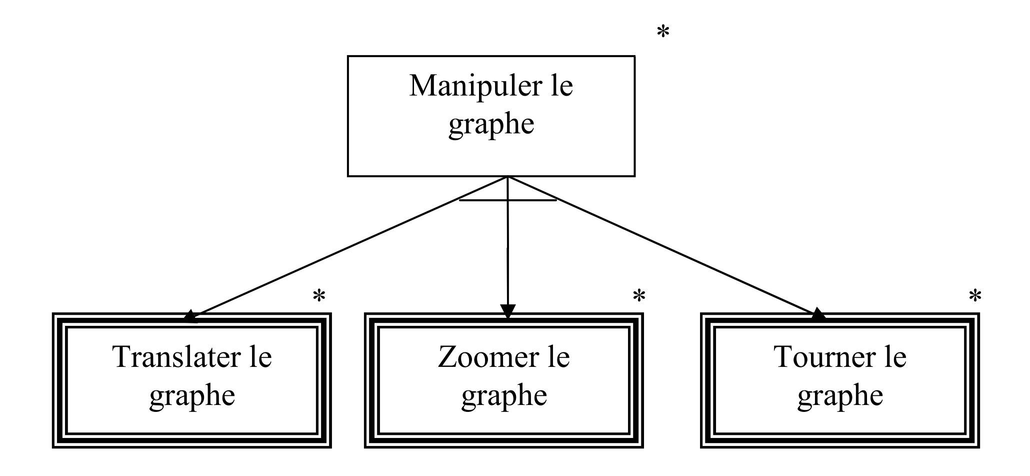


Figure 38 : Tâche manipuler le graphe

Travailler sur le graphe pouvait également consister à l'éditer. Comme le montre Figure 39, l'édition de graphe se décompose en l'extraction ou l'insertion d'un nœud ainsi que l'ajout ou la suppression d'une arête

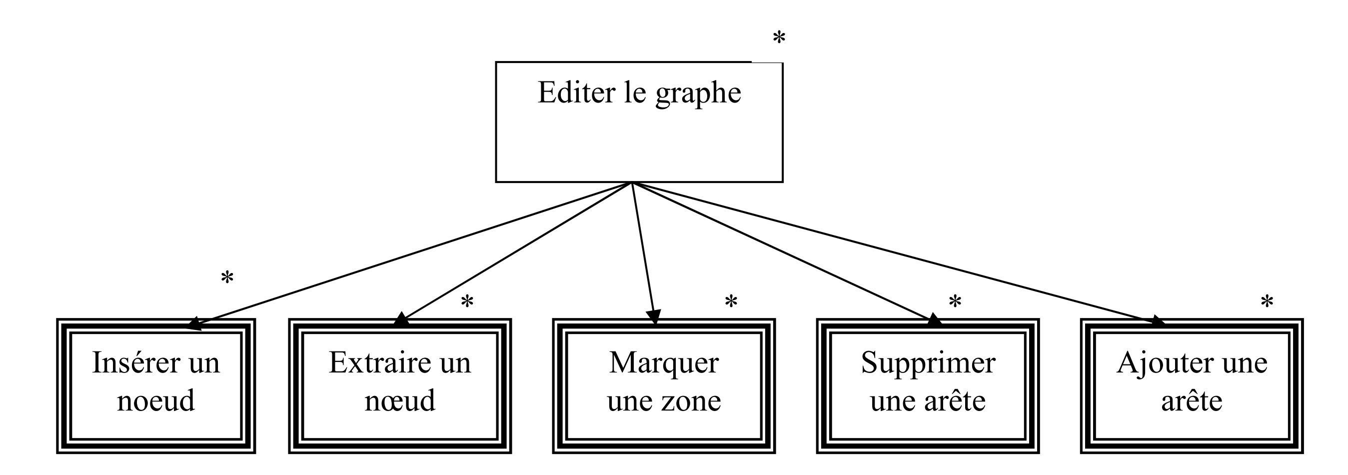


Figure 39 : Tâche Editer le graphe

A la Figure 36, nous distinguons la grappe de tâches liées à un graphe ce celle liés à un sous-graphe. La Figure 40 montre que lorsque l'on travaille sur un sous-graphe, nous pouvons soit le copier, couper, coller, soit le regrouper (clusteriser ce sous-graphe) ou le voir la zone sélectionnée en 3D.

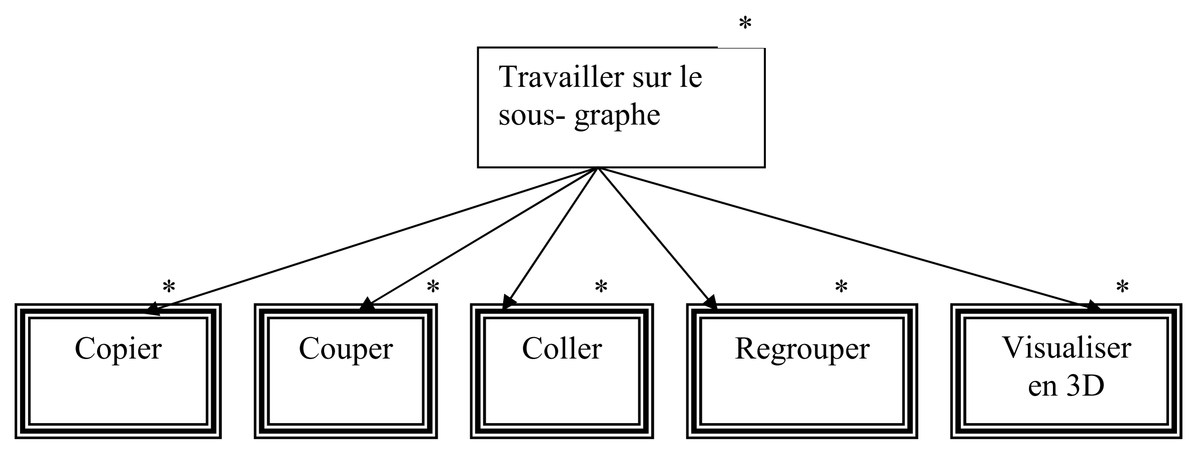


Figure 40 : Tâche travailler sur le sous –graphe

1. 2 Techniques d'interaction conçues

Nous présentons les techniques d'interaction conçues pour permettre la réalisation des tâches élémentaires du paragraphe précédent. Il s'agit ici de concrétiser les tâches élémentaires abstraites par la conception de techniques ou modalités d'interaction. Le résultat de cette étape est la spécification de l'interaction. Nous commençons par l'exposé des techniques qui sont générales et que nous regroupons dans une boîte à outils notée BiToolkit, puis nous décrivons les techniques spécifiques à la manipulation de graphes et de graphes d'ARN. Aussi notre exposé considère les techniques les plus générales d'abord jusqu'aux plus spécifiques.

1. 2. 1 BiToolKit : boîte à outils bi-manuelle

Nous présentons ici les techniques générales de manipulation de fenêtres sur la table. Une fenêtre doit pouvoir être manipulée à deux mains et être pivotée pour la montrer à un collaborateur de l'autre côté de la table. Ces deux constats ont motivé le développement de techniques d'interaction que nous regroupons dans une boîte à outils BiToolKit. Ses principaux composants sont :

- Une fenêtre graphique Window2M
- Panel
- Boutons
- Canvas

Parmi les composants graphiques, nous ne décrivons que l'interaction avec une fenêtre Window2M, interaction qui est la plus originale et repose sur une interaction à la fois à une main et à deux mains.

Interaction avec une main (un jeton) :

Déplacement :
Si le jeton est placé dans la barre des titres, la fenêtre devient active. La fenêtre est alors attachée au jeton. Le déplacement du jeton implique un déplacement de la fenêtre de la même manière. Cette manipulation est similaire au déplacement d'une fenêtre dans les interfaces graphiques standards comme le montre la Figure 41.

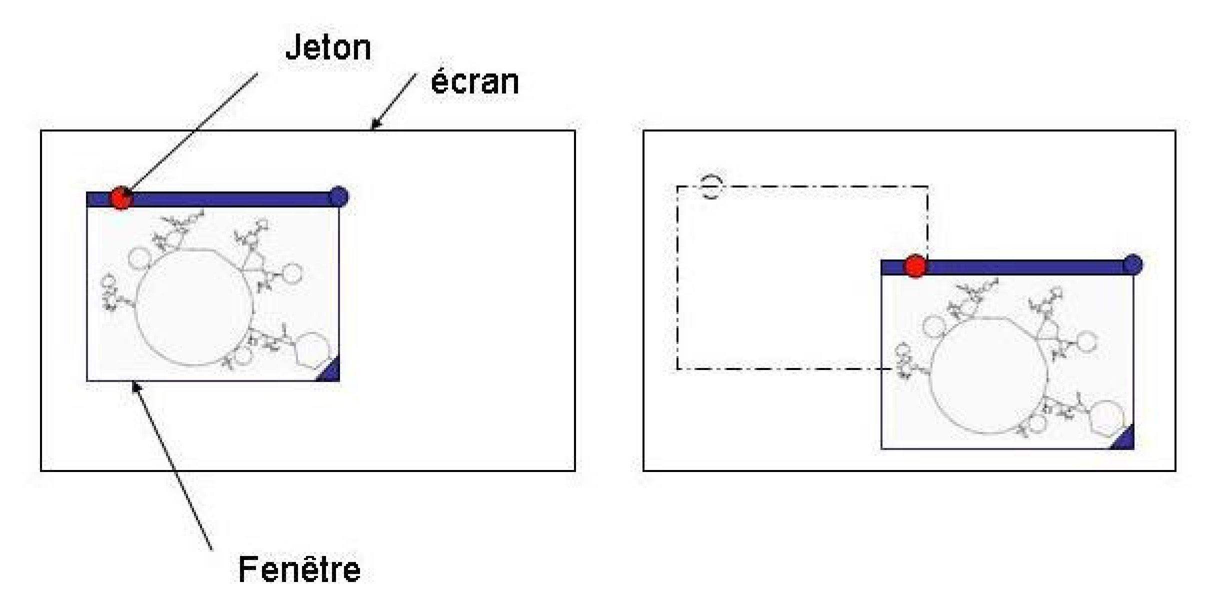


Figure 41 : Déplacement a l'aide d'un jeton d'un fenêtre Window2M

Redimensionnement:

Si le jeton est sur l'un des bords de la fenêtre, celui-ci permet de redimensionner la fenêtre. Les redimensionnements côté gauche, droit, bas et en diagonale ont été développés.

Rotation/Zoom :

L'originalité de notre fenêtre est de pouvoir faire naturellement une rotation et un zoom de la fenêtre avec une seule main comme le montre la Figure 42. Si le jeton est sur le coin spécial de la fenêtre, c'est-à-dire en haut à droit de la fenêtre, alors il est possible de faire :

- Tourner la fenêtre : le centre de rotation est défini par la croix noir en haut à gauche de la fenêtre.
- Zoomer la fenêtre : la distance entre la croix et le coin spécial de la fenêtre donne le facteur de zoom.

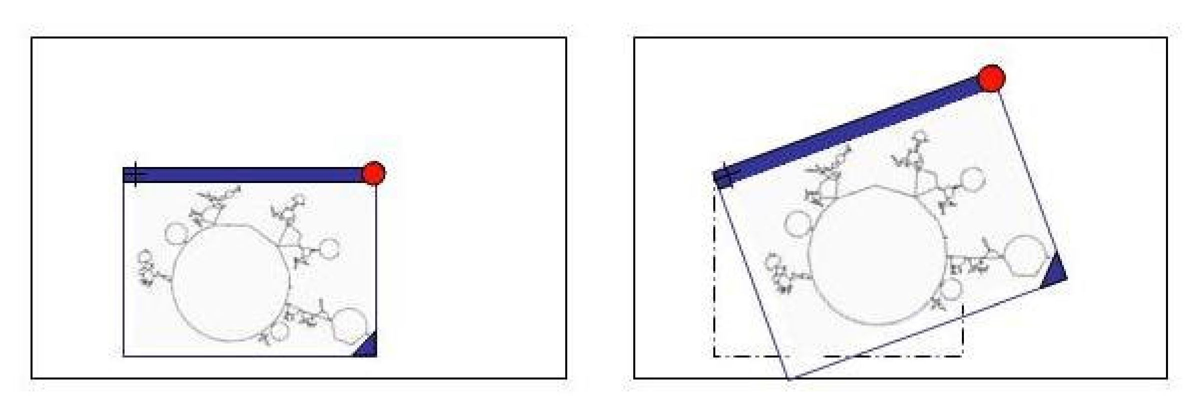


Figure 42 : Rotation et zoom d'une fenêtre window2M à l'aide d'un jeton

Nous décrivons maintenant les manipulations d'une fenêtre Window2M à deux mains.

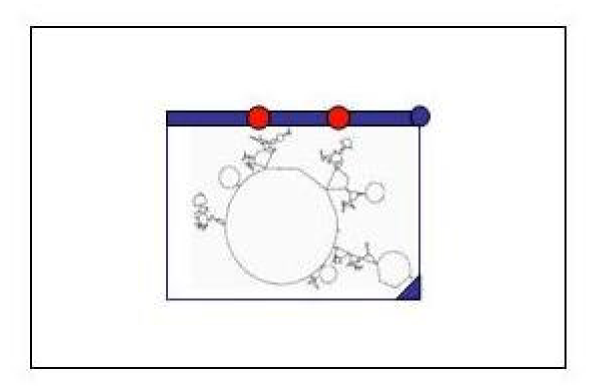
Interaction à deux mains (Deux jetons) :

Déplacement/Rotation/Zoom :

Si il existe deux jetons dans la barre de titre, la distance entre les deux jetons définit le facteur de zoom. Si on déplace un jeton, l'autre jeton joue le rôle de centre de rotation. Le jeton déplacé définit l'angle entre son ancienne et nouvelle position. Si les deux jetons sont manipulés, les micro-rotations ont pour conséquence une translation, mais l'événement qui est reçu par la fenêtre est Rotation-Zoom.

Description dans l'espace 2M de **Rotation-Zoom**: - Continu-continu. - Fusion syntaxique pour obtenir la tâche Rotation-Zoom. - Séquentiel (novice), Concomitant (expert).

Description dans l'espace 2M de **Déplacement par micro-rotations**: - Continu-continu. - Fusion syntaxique pour obtenir la tâche déplacement. - Parallèle.



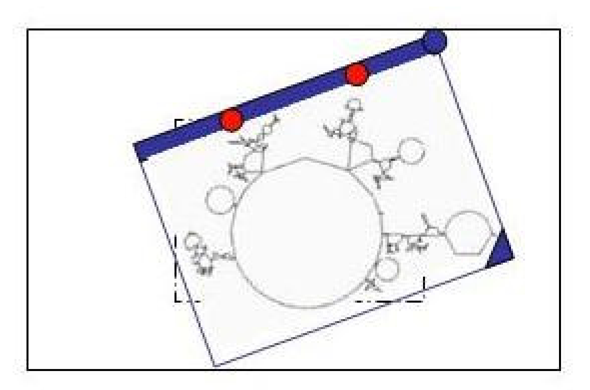


Figure 43 : Rotation et zoom de la fenêtre à l'aide des deux mains

Déplacement et Redimensionnement :

Si un jeton est dans la barre de titre et l'autre jeton sur un bord de la fenêtre. La cohérence entre les deux jetons et la fenêtre est maintenue. Il y a donc translation et redimensionnement en même temps comme le montre la Figure 43.

Description dans l'espace 2M de **Déplacement - Redimentionnement**: - Continu-continu. - Fusion syntaxique pour obtenir la tâche Déplacement-Redimensionnement - Séquentiel (novice), Concomitant (expert).

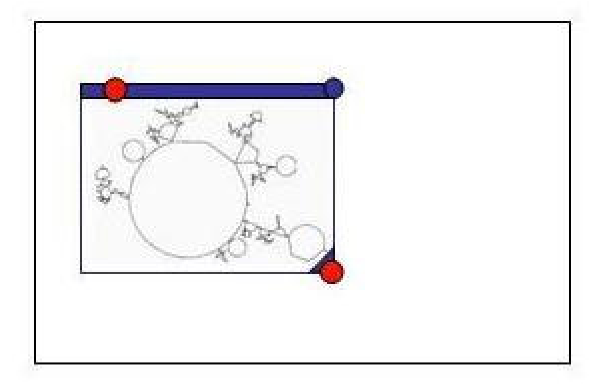
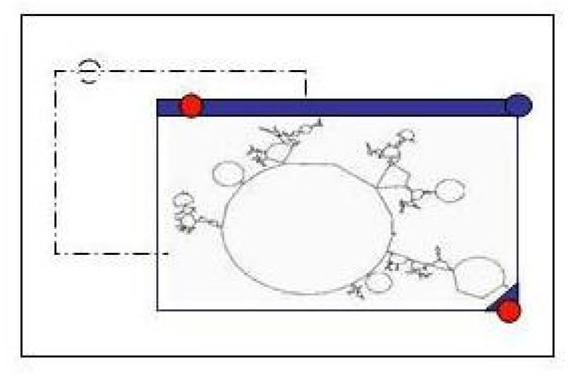



Figure 44 : Déplacement et redimensionnement de la fenêtre à l'aide des deux mains

Nous venons d'exposer les techniques d'interaction conçues pour manipuler une fenêtre Window2M. Ces techniques à une et deux mains sont générales et constituent notre boîte à outils BiToolkit entièrement développée. Nous présentons maintenant les techniques spécifiques à la manipulation d'un graphe puis à l'édition d'un graphe d'ARN.

1. 2. 2 Fonctionnalité de navGraphe

Déplacement/Zoom/Rotate :

Nous décrivons les techniques qui permettent de déplacer le graphe, de le zoomer et d'appliquer une rotation. Pour la translation, un seul jeton suffit. Le zoom et la rotation impliquent l'usage de deux jetons. Le zoom se focalise sur la zone entre les deux jetons pour maintenir la cohérence de la position relative des jetons dans le graphe. Une manipulation particulière des deux jetons permet de réaliser une translation du graphe. Cependant celle-ci ne génère pas un événement déplacement mais bien Rotation-Zoom : la translation est le résultat d'une suite de micro-rotations. Nous appliquons ici le même modèle que lors du déplacement d'une fenêtre Window2M par micro-rotations.

Description dans l'espace 2M de **la translation avec deux jetons** :
- Continu-continu.
- Fusion syntaxique pour obtenir la tâche Rotation-Zoom
- Parallèle.

Modèle de force :

Les techniques décrites dans ce paragraphe permettent de déformer le graphe mais aussi de l'éditer. Elles sont encore plus spécifiques que les précédentes puisque dépendantes du fait que le graphe manipulé est un graphe d'ARN. Les techniques reposent sur la métaphore d'un champ de force associé à un jeton, la force transformant ou modifiant le graphe.

La première technique est la déformation du graphe.

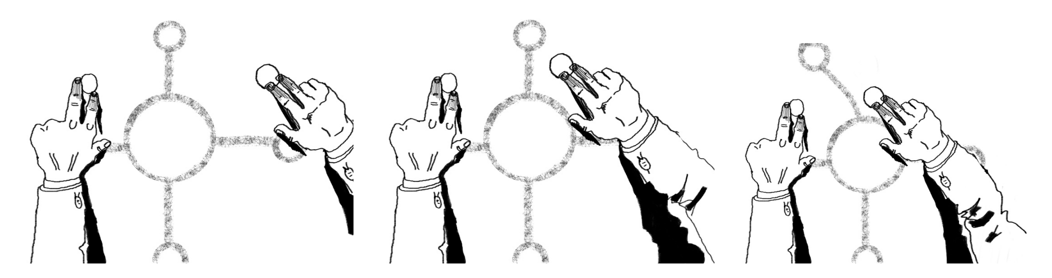


Figure 45 : Déformation du graphe grâce au modèle de force

Ici, le champ de force associé au jeton permet de déformer le graphe en pliant une des branches du graphe. Pour aboutir à la déformation voulue, l'apport des deux mains est considérable [Buxton 05].

Description dans l'espace 2M **de la déformation par le modèle de force**:
- Continu-continu
- Fusion sémantique
- Concomitant

La seconde technique est la reconstruction d'hélice d'ARN.

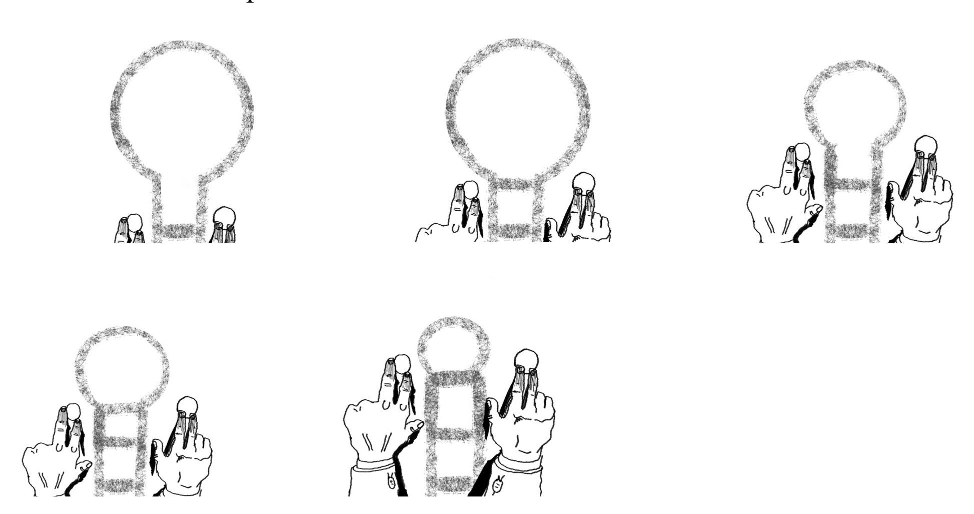


Figure 46 : Reconstruction d'hélice d'ARN

Description dans l'espace 2M **de la reconstruction d'hélice utilisant le modèle de force**
- Continu-continu
- Fusion sémantique
- Parallèle

Ici, clairement, c'est parce que nous utilisons les deux mains en même temps que la reconstruction sémantique utilisant les particularités du graphe est possible.

Ces techniques basées sur le modèle de force ont été conçues grâce à l'espace 2M, puisque nous avions décidé d'explorer les cas de fusion sémantique avec actions en parallèle. Le point dans l'espace figé, nous avons réfléchi à l'usage le plus naturel d'une telle technique pour la tâche d'édition. Ces techniques n'ont pas pu dans le temps de l'année de M2Rech être développées.

Ces techniques demandent d'approfondir Tulip (l'outil de visualisation de graphes). De plus les données manipulées étant très importantes, l'aspect algorithmique ne pourra être ignoré. En effet à chaque modification du graphe, nous nous devons de maintenir les contraintes esthétiques qui ont une complexité algorithmique importante. Ces raisons nous amènent à projeter le modèle de force comme sujet de stage jusqu'à septembre.

Même si ces techniques d'interaction n'ont pas pu être codées, elles nous semblent les plus pertinentes car elle semblent naturelles et prennent en compte la structure particulière des graphes d'ARN pour qui cette manipulation est justifiée. La conception de ces techniques a aussi permis d'illustrer le pouvoir génératif de notre espace 2M.

Sélection d'une zone :

L'utilisateur peut définir une zone de sélection du graphe. Pour cela elle/il manipule une fenêtre semi-transparente. Cette fenêtre est de type Window2M et les techniques d'interaction correspondantes sont présentées au paragraphe 1.2.1. Cette zone de sélection peut concourir à la réalisation des tâches de la Figure 40.

- **Visualiser en 3D** : l'utilisateur sélectionne une zone du graphe grâce à la fenêtre de sélection. Ce sous-graphe sélectionné est représenté sur un autre écran en 3 dimensions. Cette fonctionnalité est très attendue par les biologistes car les informations d'une molécule d'ARN sont différentes sur une vue 2D ou sur une vue 3D. Pouvoir manipuler en même temps ces deux vues semble très pertinent pour le travail des biologistes.
- **Couper/ Copier / Coller :** L'utilisateur peut sélectionner une partie du graphe. Cette partie du graphe peut être coupé ou copié afin d'être recollé à un autre endroit.
- **Regrouper / Clusteriser :** L'ensemble du graphe sélectionné par l'utilisateur peut être regroupé en un cluster afin par exemple d'être annoté ou de pouvoir se focaliser sur le reste du graphe.

Tandis que la définition d'une zone par manipulation d'une fenêtre est possible, nous n'avons pas développé la réalisation des tâches ci-dessus.

Nous venons d'exposer les techniques d'interaction conçues en les localisant dans notre espace 2M. Toutes les techniques conçues n'ont pas été développées par faute de temps. Dans ce contexte, nous savions que le code développé devait être très documenté et structuré afin d'en faciliter les modifications et extensions. A ces fins, nous avons conçu notre système de façon très modulaire grâce à l'application d'un modèle d'architecture. Ce point fait l'objet du paragraphe suivant puis nous exposons les aspects particuliers de codage qui nous semblent intéressants de souligner.



Figure 47 : Table augmentée [Berard 99] appliquée au domaine de la biologie

2 Conception logicielle

Ce paragraphe décrit l'architecture logicielle du système développé. Afin de garantir une forte modularité, et donc portabilité et modifiabilité du code, pour des extensions futures du système et en particulier l'ajout de nouvelles techniques d'interaction, nous avons appliqué le modèle d'architecture logicielle conceptuelle PAC-Amodeus. Dans le paragraphe suivant, nous rappelons le modèle PAC-Amodeus en détaillant le rôle fonctionnel de chacun de ses composants logiciels, puis nous décrivons l'architecture logicielle de notre système selon ce modèle. Nous étudions ensuite le développement de techniques d'interaction bi-manuelle au sein de l'architecture.

2. 1 Modèle d'architecture PAC-Amodeus

Un principe commun à tous les modèles d'architecture logicielle pour des systèmes interactifs réside dans la séparation de la partie logicielle gérant l'interaction Utilisateur-Système, du noyau fonctionnel, c'est-à-dire la partie logicielle maintenant les concepts et fonctions propres au domaine d'application. En partant de cette décomposition en deux briques essentielles, le modèle Seeheim [Pfaff 92] puis le modèle ARCH [Pfaff 92] proposent une décomposition plus fine du logiciel. Le modèle Arch préconise 5 composants logiciels, le Contrôleur de dialogue étant le composant principal. Le modèle PAC-Amodeus [Nigay 94] de la Figure 48 repose sur la décomposition logicielle d'Arch et affine le Contrôleur de Dialogue en termes d'agents logiciels PAC.

Le noyau Fonctionnel (NF) comprend les routines et données informatiques qui sont propres au domaine d'application. Il regroupe les objets du domaine. L'Adaptateur de Noyau Fonctionnel (ANF) réalise l'interface logicielle entre le NF et le Contrôleur de Dialogue (CD).

Le CD est chargé de mettre en correspondance les concepts manipulés par l'utilisateur dans l'interface avec les concepts informatiques manipulés par le NF. Il est également chargé de gérer l'enchaînement des tâches.

Le deuxième pilier de l'arche, comprend en son sommet le composant Présentation Logique (LP). Il exprime de manière abstraite le comportement des objets interactionnels requis par l'interface. Le composant LP permet donc de rendre le CD indépendant de la boîte à outils du composant Présentation (P) implémentant les objets d'interaction de l'interface (fenêtre, boutons, etc.), de la même manière que le CD est rendu indépendant du noyau fonctionnel par le biais de l'ANF. Le composant LP permet la réalisation d'agrégats d'objets d'interaction dans le cas où ceux proposés par la boîte à outils ne satisfont pas les besoins du CD. Enfin, le dernier composant est la Présentation (P). Le composant P regroupe les éléments logiciels et matériels de la plate-forme : la boîte à outils, les pilotes des dispositifs, etc. C'est donc au niveau du composant P que les actions physiques de l'utilisateur sont captées : déplacement de la souris, touches de clavier appuyées, etc. Elles sont ensuite interprétées et transmise au CD via le composant LP. Le CD vérifie alors la cohérence de l'action initiée par l'utilisateur par rapport au contexte courant d'interaction et, via l'ANF, fait appel aux fonctionnalités du NF pour effectuer la commande. Si l'action implique un retour d'information du niveau sémantique à l'utilisateur, celui-ci est généré par le NF, envoyé via l'ANF au CD qui l'adapte pour le rendre perceptible par l'utilisateur au niveau du composant P via le composant LP.

Le contrôleur de dialogue est peuplée d'agents PAC, agents à trois facettes : Présentation, Abstraction et Contrôle :

- La présentation définit le comportement perceptible de l'agent pour agent humain. Elle concerne à la fois les entrées et les sorties c'est-à-dire les modalités d'action accessibles à l'utilisateur et la restitution perceptible.
- L'abstraction, avec ses fonctions et ses attributs internes, définit la compétence de l'agent indépendamment des considérations de présentation. Elle constitue, au sens de Seeheim, le noyau fonctionnel de l'agent. Le Contrôle a un double rôle :
  - Il sert de pont entre les facettes présentation et abstraction de l'agent.
  - Il gère les relations avec les autres agents PAC. C'est par leur partie contrôle que deux agents PAC communiquent.

Ce modèle d'architecture constitue un cadre conceptuel adapté à la visualisation de données, à l'interaction multimodale en entrée [Nigay 94], en sortie [Vernier 01] mais aussi aux interactions mixtes combinant les mondes réel et numérique [Renevier 04].

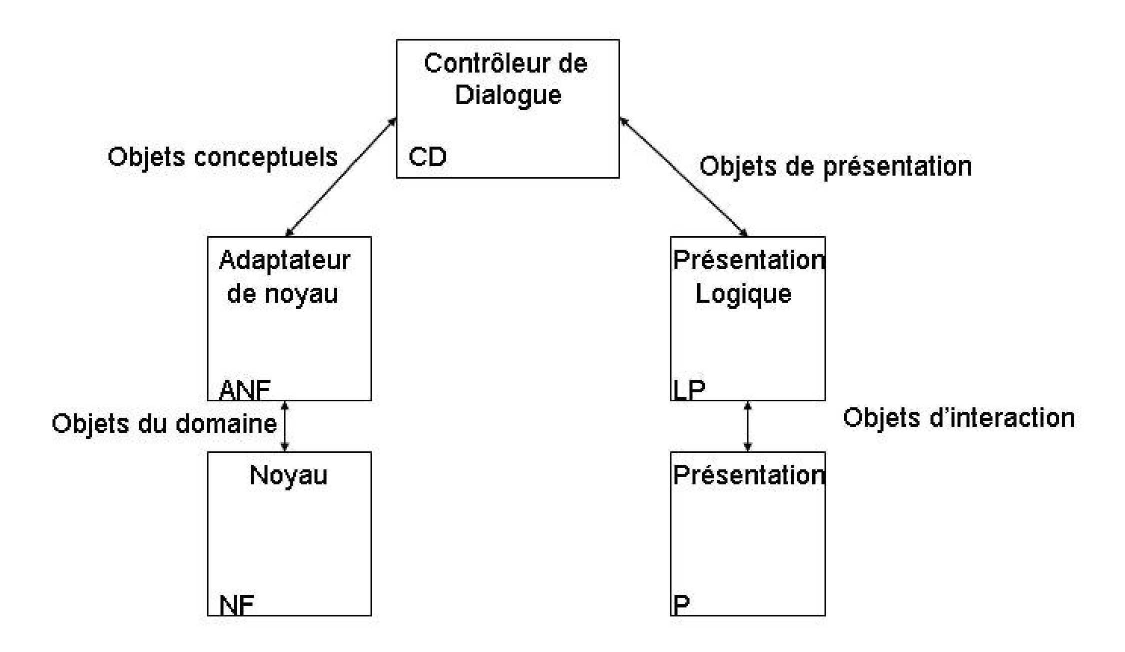


Figure 48 : Architecture PAC-Amodeus

2. 2 Architecture du système NavGraphe

L'architecture de notre application vérifie le modèle PAC-Amodeus, comme le montre la Figure 49. Le NF contient l'ensemble des fichiers correspondant aux graphes manipulables par le système. Nous utilisons une boîte à outils de visualisation de graphes, Tulip [Auber 01]. Plus qu'une boîte à outils dédiée uniquement au graphique, elle comprend des fonctions de transformation de format de graphes. Aussi les fonctions de Tulip qui typiquement devrait être exploitées uniquement au niveau du Composant P sont aussi utilisées dans l'ANF pour transformer la structure du graphe du NF en une structure de graphe Tulip.

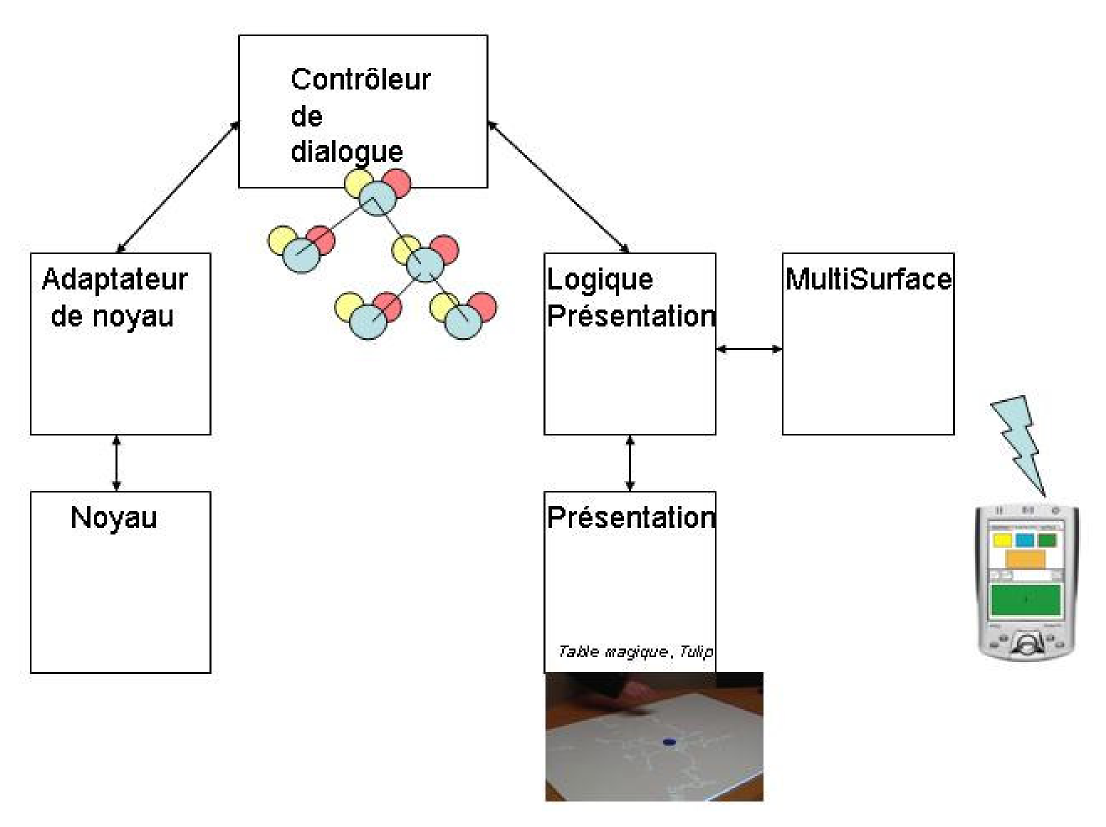


Figure 49 : Architecture de l'application qui suit le modèle PAC-Amodeus

Cette structure est maintenue par l'abstraction de l'agent Graphe dans le CD. Cette structure de graphe est notée « SuperGraph » dans Tulip. Dans le cas idéal et pour une indépendance du CD vis-à-vis des boîtes à outils, le composant LP devrait traduire les demandes de présentation du CD en appel de fonctions des boîtes à outils utilisées contenues dans le composant P : Tulip pour afficher le graphe et notre boîte à outils BiToolkit pour les fenêtres manipulables à deux mains. Néanmoins dans notre développement, le CD est dépendant de la boîte à outils Tulip. Enfin le composant P contient aussi la boîte à outils TclVision [Bérard 99] pour le suivi de jetons. Les événements reçus de cette boîte à outils sont décrits en Annexe de ce mémoire.

Si nous rajoutons des surfaces d'interaction comme les murs, le composant LP gérera l'ensemble de ces surfaces. Aussi si un ordinateur de poche sert de télécommande pour configurer l'environnement informationnel composé de plusieurs surfaces, nous rajoutons une arche au sein de l'architecture initiale dont le point de contact est le composant LP. Cette nouvelle arche structure le code de la meta-interface (meta-ui) que constitue la télécommande de configuration.

Au sein du CD, la hiérarchie d'agents PAC est à quatre niveaux, comme montrée à la Figure 50. Nous avons appliqué la règle de construction intitulée ciment syntaxique : "Si la spécification d'une commande implique des actions distribuées sur plusieurs agents, ceux-ci doivent être placés sous le contrôle d'un agent qui cimente les actions réparties en une commande." [Nigay 94]. Ainsi, nous avons un agent Racine qui est au sommet de la hiérarchie de tous les agents. Ses deux agents fils sont l'agent Requête et l'agent Ciment :

- L'agent requête est pour l'instant vide mais permettra d'interroger des bases de données biologiques dans le futur.
- L'agent Ciment assure la communication entre ces trois agents fils : Historique, Commande et Graphes afin de les rendre indépendants.
- L'agent Historique sera utilisé dans le futur pour mémoriser les commandes effectuées afin de pouvoir revenir dans un état précédent en cas d'erreur.

L'agent Commande gère différents modes : mode Manipulation, mode Edition, mode Force. L'agent Graphes gère l'ensemble des graphes. C'est donc le père de tous les agents Graphe. L'agent Graphe est responsable de la gestion du graphe. Sa facette abstraction possède la structure « SuperGraph » de Tulip tandis que sa facette présentation possède la structure GlGraph (représentation graphique du graphe en OpenGl sous Tulip) et les caractéristiques de la caméra qui regarde le graphe.

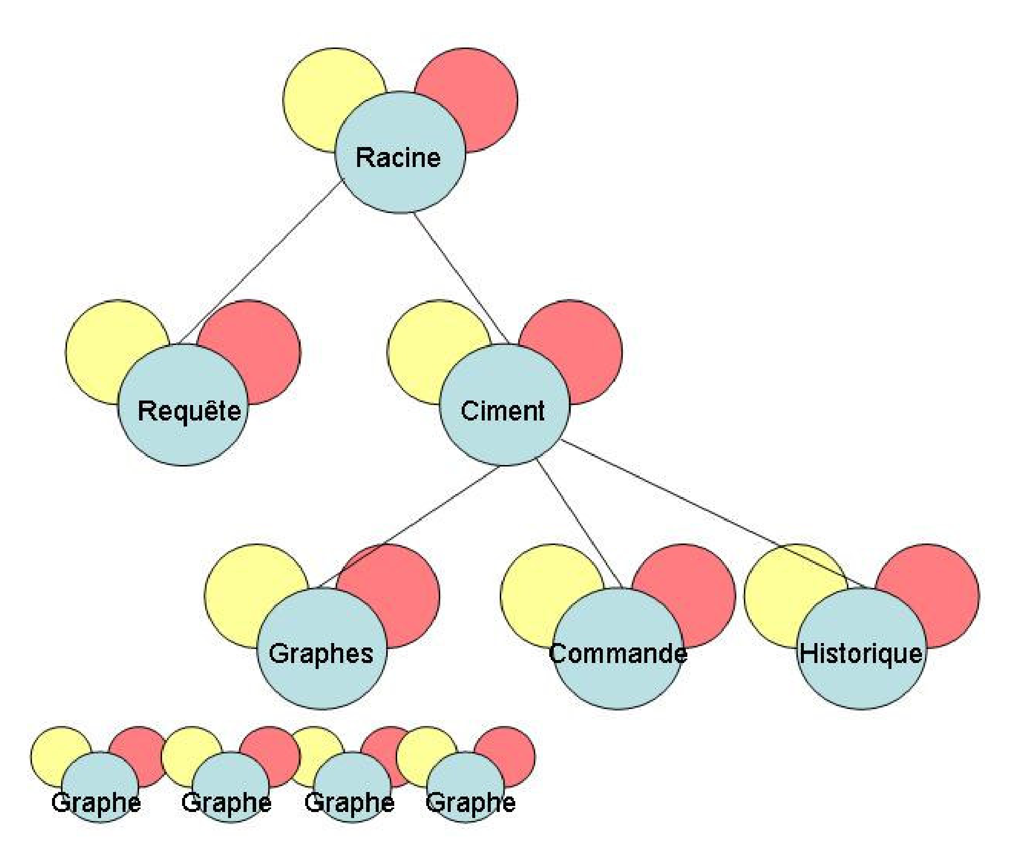


Figure 50 : Hiérarchie d'agents PAC du contrôleur de dialogue

2. 3 Solution architecturale générique pour l'interaction bi-manuelle

Les différents types de fusion de l'axe Dépendance de notre espace de conception 2M ont été identifiés dans le contexte de l'interaction multimodale. Ces types de fusion ont été localisés au sein du modèle PAC-Amodeus dans [Nigay 94]. Ce résultat est donc directement applicable pour la conception logicielle d'interaction bi-manuelle. Nous décrivons le lieu de ces fusions au sein du modèle.

2. 3. 1 Fusion lexicale

Rappelons que la fusion lexicale est la fusion de bas niveau. Celle-ci est généralement réalisée par la boîte à outils elle-même. Dans TCLVision il n'y a pas de fusion lexicale pour les événements liés aux jetons. Lorsque nous recevons un événement lié à un jeton, nous ne recevons pas d'information sur la position des autres jetons sur la table. Nous pourrions étendre cette boîte à outils au sein du composant Présentation. Nous proposons de réaliser la fusion lexicale entre les composants P et LP à l'aide d'une carte des jetons maintenus dans le composant Présentation. Cette carte contient à tout instant les propriétés de tous les jetons sur la table. Lorsqu'un nouveau jeton est apparu, il met à jour cette carte.

2. 3. 2 Fusion syntaxique

La fusion syntaxique correspond à la fusion des paramètres. Par exemple, pour le graphe nous avons une fonction Zoom (Token* t1, Token* t2). Cette fonction prend en paramètre deux jetons. La distance entre ces deux jetons fournira le facteur de zoom. Nous sommes typiquement dans un cas de fusion syntaxique, ou le jeton associé à chaque main est un paramètre d'une fonction. Cette fusion syntaxique s'effectue entre les composants LP et CD.

2. 3. 3 Fusion sémantique

La fusion sémantique est la fusion des tâches. Cette fusion a lieu dans le contrôleur de dialogue. Dans le cas de la reconstruction d'hélice d'ARN ou construction d'arêtes par compression du graphe (modèle de force) à l'aide de deux jetons, si le contrôleur de dialogue reçoit dans une fenêtre temporelle arbitrairement petite deux commandes « poussée du graphe par la gauche » et « poussée du graphe par la droite » ces deux tâches doivent être fusionnées afin de créer la commande « Compresser le graphe ».

Nous venons d'expliquer la conception logicielle de notre système NavGraphe selon le modèle PAC-Amodeus. Nous exposons maintenant les aspects de codage de notre système.

3 Développement

Le système est développé selon une technologie orientée objet. Le langage utilisé est le C++ pour plusieurs raisons :

- Compatibilité avec tulip écrit en C++.
- Facilité pour intégrer openGl et GLUT.
- Soucis de performance car les graphes manipulés peuvent être de grandes tailles.
- Réutilisabilité et maintenance du code, car les concepteurs susceptibles de poursuivre ce travail sont des programmeurs C++.

La partie graphique du système s'appuie sur la bibliothèque GLUT pour simplifier le travail du développeur au niveau de la création de la fenêtre principale, de la gestion de la boucle d'affichage et de la récupération des événements souris pour le simulateur.

Pour le développement, nous travaillons sous linux afin de pouvoir modifier facilement le source de Tulip. C'était la première fois que Tulip était intégré dans un logiciel, d'habitude les concepteurs développent des plugins qui sont chargés au démarrage de Tulip. Notre approche

était donc tout à fait novatrice de considérer Tulip comme une boîte à outils. Nous nous sommes donc préparés à modifier les sources de Tulip pour réagir à des problèmes imprévus. Il s'avère qu'il est beaucoup plus facile de modifier le source de Tulip sous Linux. La raison principale est que l'interface de Tulip utilise la librairie QT qui est payante sous windows alors qu'elle est gratuite sous linux.

L'environnement de programmation utilisé est eclipse avec le plugin CDT qui gère les projets C et C++. Il a l'avantage de créer un makefile de bonne qualité automatiquement. Comme second avantage Eclipse est multi-plaforme car écrit en java. Si dans l'avenir, nous souhaitons porter notre logiciel sous windows ou mac, notre travail sera plus aisé. Eclipse était donc utilisé pour la gestion des makefiles et la création de classes.

Pour le développement pure, nous utilisions emacs pour les raisons suivantes :

- Il est plus rapide que Eclipse qui est très consommateur de ressources.
- Le code est beaucoup plus lisible grâce à une meilleure gestion des couleurs.
- L'indentation est beaucoup mieux gérée ce qui est important pour la relecture de code.
- En ouvrant deux consoles en même temps, il est possible avec emacs de voir en même temps le fichier .h et le fichier .cpp.

Nous souhaitons enfin insister sur toutes les techniques de développement que nous nous sommes imposées afin de garantir une forte modularité et propreté du code. En effet nous savions que les techniques d'interaction ne pouvaient être développées dans le temps du M2rech et nous avons préféré développer un code réutilisable sans toutes les techniques d'interaction plutôt qu'un démonstrateur jetable.

Le travail de Nicolas Gandin portait l'année dernière au sein de l'équipe sur la visualisation de grands graphes. Il avait l'avantage de montrer les possibilités de Tulip et la compatibilité avec TclVision. Malheureusement, le manque d'architecture et surtout le manque complet de documentation a rendu la réutilisation de son code très difficile voir impossible.

A ce jour le code du système NavGraphe comprend 10 000 lignes de code pure (déclaré par le logiciel sloccount) et 10000 lignes de commentaires répartis dans plus de 60 classes (120 fichiers). Nous développons dans les paragraphes suivants les éléments qui contribuent à la qualité du code produit à ce jour.

3. 1. 1 Architecture globale : PAC-Amodeus

Le nom des classes ainsi que la structure de mon projet suit l'architecture PAC-Amodeus. Nous retrouvons par exemple :

- un répertoire pour chaque composant ARCH du modèle PAC-Amodeus.
- un répertoire pour chaque agent du contrôleur de dialogue.

Chaque agent est constitué de trois classes <nomAgentA>, <nomAgentC>, <nomAgentP> respectivement pour la facette abstraction, la facette contrôle et la facette présentation de l'agent.

Chaque facette d'un agent hérite de la facette élémentaire qui lui correspond. Par exemple monAgentA hérite de facetteElementaireAbstraction. Ces facettes élémentaires imposent les règles de communication entre les différentes facettes des agents et les agents eux-mêmes.

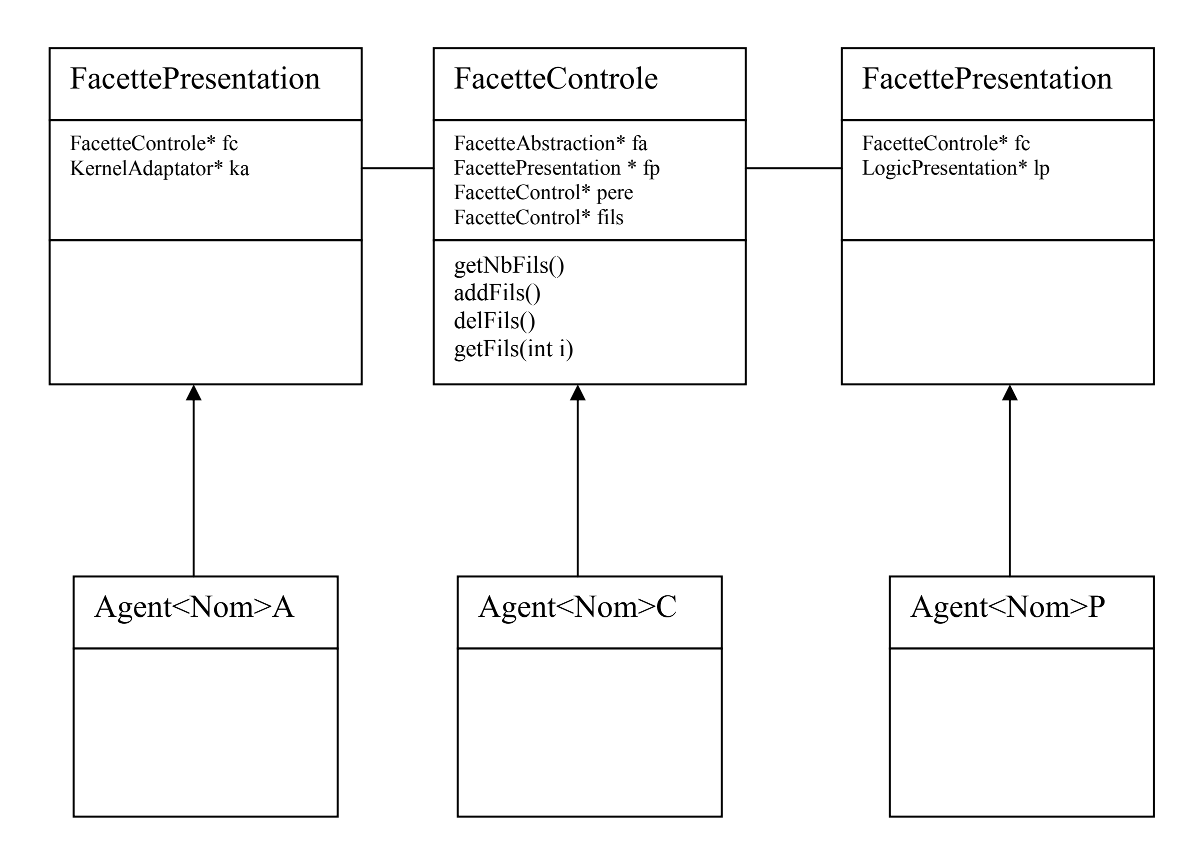


Figure 51 : Modèle UML d'un agent PAC

Cette règle permet de facilement réutiliser les fonctions au sein du code. Par exemple, la méthode qui importe un graphe Tulip se trouve dans l'ANF. La structure du graphe Tulip (SuperGraph) se trouve dans la facette abstraction de l'agent Graphe. Sa représentation en OpenGl (GlGraph) se trouve dans la facette présentation de l'agent Graphe.

De plus, j'ai sauvegardé mon projet au moment où l'architecture PAC-Amodeus a été élaborée. Ainsi, nous disposons d'une bonne base pour le développement de toute application multimédia qui repose sur l'architecture PAC-Amodeus. La structure vide PAC-Amodeus est développée et réutilisable.

3. 1. 2 Patrons de conception

Pendant cette année de M2rech, j'ai découvert les patrons de conception (design pattern) [Gamma 95]. Je me suis alors attaché à les appliquer. Dans le domaine de l'analyse et de la conception orientée-objet, un patron de conception est une manière de construire la structure d'une classe. Plus généralement, un motif de conception est un document qui décrit une solution générale à un problème qui revient souvent. Les motifs sont basés sur des expériences passées avec les mêmes structures. Les patrons de conception sont ainsi des techniques permettant d'augmenter la productivité en adoptant certaines structures établies et réutilisables et de faciliter la communication entre développeurs.

3. 1. 3 Optimisations

L'un des problèmes rencontrés avec le démonstrateur de Nicolas Gandin est la latence. Un effort particulier d'optimisation a donc dû être nécessaire pour réduire cette latence et rendre le système utilisable. Tulip manipule des graphes qui ont plusieurs milliers de nœuds. Il faut donc minimiser le nombre d'appels à la fonction d'affichage de Tulip. Dans l'ancien prototype, pour chaque événement provenant de la table, un appel à Tulip était effectué. Comme il faut 1/100 de seconde à Tulip pour afficher le graphe, pendant ce temps plusieurs événements jeton s'accumulent dans la queue d'événements et seront traités un par un dès que Tulip aura fini d'afficher. Il semble évident que les événements jetons intermédiaires devraient être ignorées afin d'améliorer la réactivité du système (c'est ce qui se passe pour les événements souris dans les interfaces standards).

Pour cela, nous avons proposé d'ignorer les événements intermédiaires en utilisant un objet partagé. Le programme est donc composé en trois parties :

- la boucle idle de GLUT qui est appelée en permanence pour l'affichage.
- Un thread qui écoute les événements jetons sur la socket.
- Un objet partagé, en l'occurrence une table indexée par l'identifiant des jetons (map) qui maintient en permanence l'état des différents jetons sur la table. Cette table correspond à l'implémentation de la notion de carte des jetons que nous avons exposé au paragraphe 2.3.1 pour la réalisation de la fusion lexicale.

Du côté réseau, dès qu'un événement jeton arrive sur la socket, le thread met à jour la carte partagée et déclare qu'elle a été modifiée. A chaque passage dans la boucle d'affichage idle (c'est-à-dire dès que le programme a le temps pour s'afficher) un appel à la carte partagée est effectué. Si celle-ci a été modifiée, alors on demande à Tulip de se réafficher, sinon on ne fait rien. Ainsi, si durant l'affichage de Tulip, il y a eu 100 événements jetons (par exemple un déplacement continu de la part de l'utilisateur), seul le dernier événement sera considéré. Nous maintenons en temps réel, la cohérence entre la position des jetons et l'état de la scène affichée.

Une autre technique d'optimisation de l'affichage a été élaborée. Le réaffichage de la scène n'implique pas forcément le réaffichage de tout le graphe. Par exemple, pour le déplacement d'un graphe, au lieu de demander à Tulip de se réafficher pour chaque position, le programme réalise une image du graphe au départ qui sera mise dans une texture. Ainsi au lieu de déplacer le graphe (et donc d'appeler la fonction d'affichage de Tulip), le programme déplace l'objet texturé par un appel à des fonctions openGl très efficaces. Donc, si le programme n'a pas réellement besoins des compétences de Tulip, il sollicite l'objet texturé pour qu'il se réaffiche. Cette technique a été implémentée pour la translation, la rotation et une partie du zoom puisque le zoom sémantique implique le réaffichage de graphe. Les résultats obtenus sont bien supérieurs au démonstrateur développé l'année dernière.

3. 1. 4 Une classe, deux fichiers

Une classe est décomposée en deux fichiers afin de séparer la définition des méthodes de la façon dont elles sont réalisées :

- Le fichier classe.h qui fournit l'entête de toutes les fonctions, les attributs de la classe et les commentaires sur les méthodes et les attributs,
- le fichier classe.cpp où sont implémentés les méthodes de la classe.

La structure de mes classes suit les règles imposées par la communauté Linux. Il convient de souligner que très peu de livre C++ explique comment faire du code propre à tel point que très souvent la notion de fichier d'entête est souvent ignorée.

3. 1. 5 Documentations

Chacune des méthodes ont été documentées. Les commentaires suivent la syntaxe Doxygen. Doxygen est un outil de génération de documentation semblable à javaDoc pour java. Il est plus puissant car il supporte plusieurs langages de programmation, plusieurs formalismes. Il montre la hiérarchie des classes par rapport à la classe courante. La documentation du système NavGraphe développé est disponible à l'adresse http://iihm.imag.fr/bailly/navgraphe/

3. 1. 6 Installation

Les difficultés rencontrées lors de l'installation de Linux et du pilote de la carte graphique nous ont incité à rédiger un manuel d'installation de ce qui est nécessaire pour développer et exécuter le système NavGraphe. Ce manuel est fourni en Annexe.

3. 1. 7 Conclusion sur le développement

Dans un souci de réutilisabilité et d'extensibilité du code, nous avons visé un code propre, rigoureux et optimisé. Cela comprend :

- un modèle architecture explicite PAC-Amodeus,
- une structure de fichiers propres,
- l'application de patrons de conception,
- une documentation complète,
- un code optimisé pour réduire la latence dans l'interaction.

4 Conclusion sur les techniques conçues et développées

Dans ce chapitre, nous avons décrit un ensemble de techniques d'interaction bi-manuelle que nous avons conçues et localisées au sein de l'espace 2M. De nombreuses autres techniques sont à concevoir en considérant d'autres points de l'espace 2M. Parmi les techniques bi-manuelles conçues, certaines sont générales comme la manipulation de fenêtres et composent notre boîte à outils BiToolkit, d'autres sont spécifiques à la manipulation d'un graphe, voir spécifique à un graphe d'ARN.

Nous avons ensuite expliqué la réalisation logicielle des techniques conçues. Nous avons opté pour un code réutilisable et extensible sans toutes les techniques d'interaction conçues, plutôt qu'un démonstrateur jetable de toutes les techniques. Ce choix a donc impliqué un très gros travail de conception logicielle et d'implémentation rigoureuse. Il est évident que faire du « code de bonne qualité» est un investissement.

Comme tout investissement, cela a coûté en temps, mais les bénéfices commencent déjà à être perceptibles. Ainsi au cours du mois de mai, nous avons encadré quatre étudiants RICM2

de l'école Polytech'Grenoble en stage d'un mois. Nous leur avons proposé un sujet en relation avec le système NavGraphe mais suffisamment indépendant du système pour distinguer les contributions de chacun. Leur sujet consistait en la gestion de plusieurs surfaces d'interaction dont la table augmentée, sur un ordinateur de poche qui sert de télécommande de configuration. Le sujet proposé et les documents en rapport avec ce stage sont fournis en annexe. Après plusieurs réunions pour figer l'architecture et les points d'ancrage avec notre système (Figure 49), les étudiants ont développé leur système en java, là encore en appliquant le modèle PAC-Amodeus. Grâce à notre approche basée sur une architecture logicielle explicite dans le code, l'intégration des deux systèmes a été immédiate.

Chapitre VII Conclusion

Nos travaux contribuent à la conception ergonomique et à la réalisation logicielle de techniques d'interaction bi-manuelle. L'originalité de l'approche adoptée réside dans l'étude de l'interaction à deux mains au regard de la multimodalité (une modalité par main) pour prendre en compte les aspects temporels et de dépendance de la multimodalité lors de la conception de techniques d'interaction bi-manuelle.

1 Synthèse de notre contribution

Notre mémoire s'organise en deux parties, l'une dédiée à la conception de l'interaction bi-manuelle et l'autre à l'application de nos contributions au cas de la manipulation de graphes biologiques. La Figure 52 résume notre démarche de travail mise en relation avec les chapitres du mémoire.

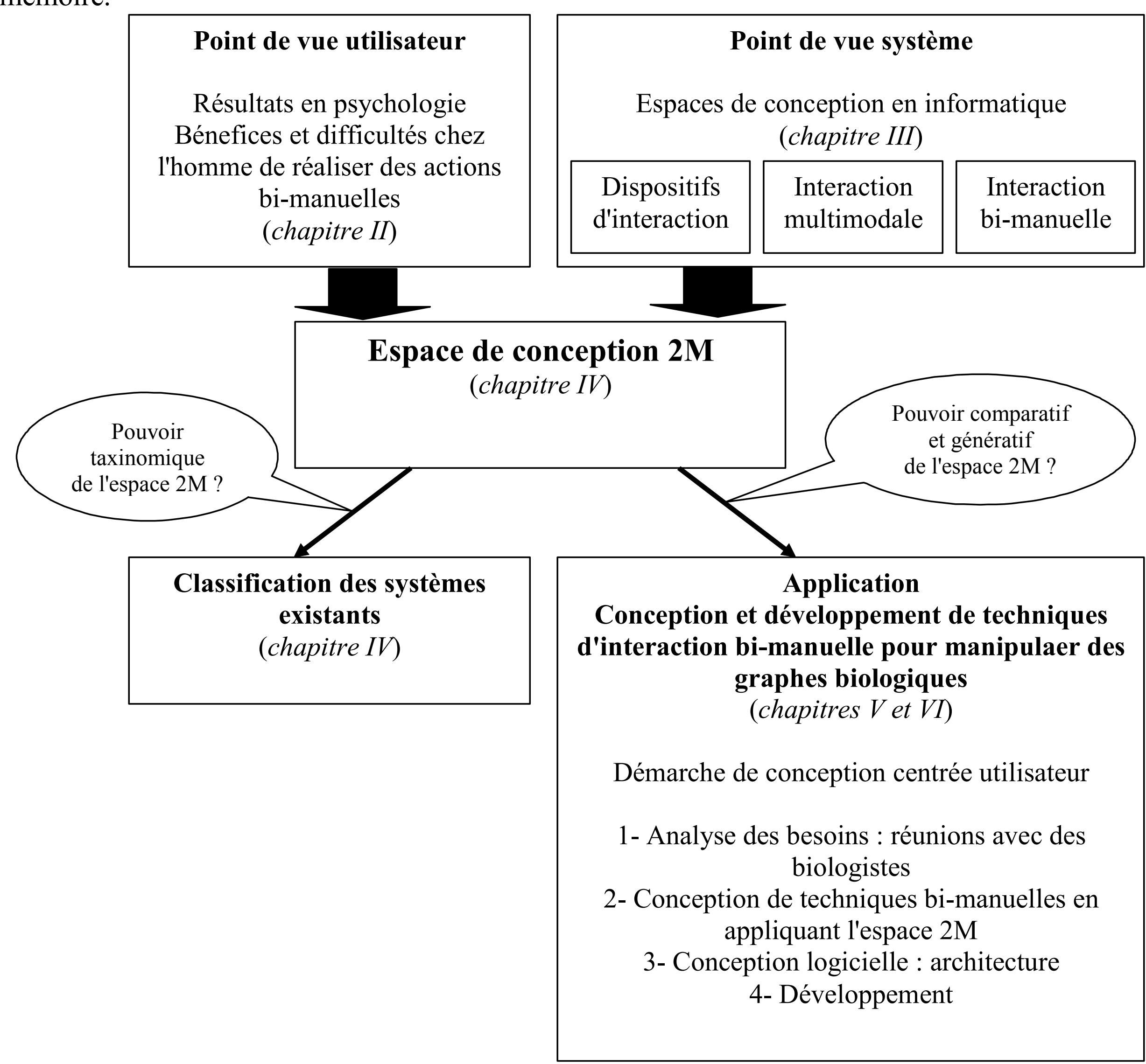


Figure 52 : Démarche de travail et chapitres du mémoire

La première partie de notre mémoire composée de trois chapitres est consacrée à la conception de l'interaction bi-manuelle.

Afin de définir un canevas intégrateur des activités de l'utilisateur et du système lors de l'interaction bi-manuelle, nous avons mené une analyse conceptuelle des résultats en psychologie (point de vue utilisateur) au chapitre II, puis une étude comparative des espaces de conception en informatique (point de vue système) au chapitre III. Parmi les espaces de conception étudiés, nous considérons ceux dédiés aux dispositifs d'entrée puisque souvent les deux mains ne tiennent pas le même dispositif, ceux dédiés à la multimodalité en entrée en considérant une modalité par main et enfin un espace de conception dédié à l'interaction bi-manuelle. Cette étude des espaces de conception nous permet de définir les bases de notre contribution mais aussi de souligner l'originalité de notre démarche qui allie interaction multimodale et interaction bi-manuelle, deux perspectives sur l'interaction qui n'avaient pas été étudiées conjointement jusqu'à aujourd'hui.

Nous concluons la partie conception de notre mémoire par l'exposé de notre contribution, l'espace de conception 2M, au chapitre IV, espace qui est dédié à l'interaction bi-manuelle et qui est issu de la capitalisation de plusieurs résultats en interaction multimodale et bi-manuelle. Notre espace 2M s'articule selon trois axes pertinents pour la conception de l'interaction bi-manuelle qui sont (1) le type d'actions effectuées par chaque main, (2) les relations temporelles entre les actions des deux mains, et enfin (3) la dépendance des actions de chaque main.

Comme pour tout espace de conception, la pertinence de l'espace 2M est étudiée au regard de son pouvoir taxinomique mais aussi de son apport lors de la phase de conception en mesurant son pouvoir comparatif d'alternatives de conception et son pouvoir génératif de nouvelles techniques. Tandis que le positionnement de nombreux systèmes existants au sein de l'espace 2M a permis de souligner son pouvoir taxinomique, montrer ses apports en phase de conception (pouvoir comparatif et génératif) est plus difficile. En effet, la validation implique en particulier d'être capable de mesurer l'impact de l'espace 2M sur la solution conçue. Un premier pas vers cette validation a été d'appliquer notre espace à la conception d'un système interactif bi-manuel, objet de la deuxième partie de notre mémoire.

La deuxième partie du mémoire vise à valider nos résultats en les mettant en œuvre dans la conception et la réalisation logicielle d'un système interactif bi-manuel. Le domaine d'application est la manipulation bi-manuelle de graphes d'ARN par des biologistes. L'approche de conception des techniques d'interaction repose sur l'étude des besoins des biologistes et notre espace 2M. Nous avons touché du doigt la difficulté de l'analyse de l'activité et de la tâche lors de la phase d'identification des besoins (chapitre 5). L'espace 2M nous a incité à systématiquement explorer l'espace solution, et en particulier l'axe dépendance nous a fortement influencé pour la conception de la technique d'interaction à deux mains en utilisant un modèle de force. Les techniques ainsi conçues (chapitre 6) ont ensuite été développées selon un modèle d'architecture et une méthode de programmation systématique. Comme il s'agit d'une plateforme logicielle que nous souhaitons étendre pour explorer plus en avant l'interaction bi-manuelle et pour l'appliquer à d'autres domaines d'application, nous avons été amené à être extrêmement rigoureux tant sur l'architecture logicielle que sur le code et la documentation.

En synthèse, nos contributions sont à la fois conceptuelles et pratiques :

- un nouvel espace de conception 2M pour l'interaction bi-manuelle, publié et présenté à la conférence UBIMOB 2005 (ACM International Conference Proceedings Series),

- une classification des systèmes bi-manuels existants,
- un nouveau système bi-manuel avec des jetons (localisés par caméra) pour la manipulation collaborative de graphes d'ARN projetés sur une table (20 000 lignes de code -code pure + commentaires- réparties dans 60 classes).

2 Perspectives

Les deux facettes de nos travaux, conception et réalisation, sont également apparentes dans les perspectives que nous donnons à cette étude. Dans ce paragraphe, nous présentons tout d'abord des perspectives à court terme avant de présenter nos perspectives à plus long terme.

2. 1 Extensions

A court terme, les extensions concernent principalement la réalisation logicielle et les évaluations à mener sur notre système.

Un premier travail à court terme consiste donc à programmer toutes les techniques d'interaction bi-manuelle présentées dans ce mémoire. Pour la technique d'interaction bi-manuelle avec modèle de force, la programmation devra aussi permettre de respecter les critères d'esthétisme de graphes des biologistes. Il sera donc indispensable d'approfondir les algorithmes et la structure utilisés dans la boîte à outils Tulip.

La réalisation des techniques achevées, nous envisageons de conduire des expérimentations avec des biologistes. Ceci sera facilité par le fait que des biologistes collaborent avec l'équipe. Le résultat de ces évaluations enrichira l'espace 2M en qualifiant l'utilisabilité de certains points de l'espace dans le contexte de la tâche considérée.

2. 2 Prolongements

Nos perspectives s'organisent selon deux axes complémentaires : le premier axe de recherche concerne la conception de techniques d'interaction bi-manuelle, tandis que le deuxième est lié à la plateforme logicielle d'interaction bi-manuelle.

Espace de conception 2M
Comme nous l'avons souligné dans le mémoire, de nombreuses études liées à l'espace 2M sont envisagées. La mesure de l'impact de l'espace 2M sur l'utilisabilité des solutions conçues nous semble intéressant d'étudier, par exemple par une étude comparative en demandant à deux équipes de concepteurs l'un utilisant l'espace 2M comme outil de conception l'autre sans de travailler sur un projet.

De plus le pouvoir comparatif de l'espace implique de qualifier chaque point de l'espace du point de vue utilisabilité. Ce travail qui nécessite le développement et l'évaluation de techniques d'interaction permettra d'établir un véritable guide de conception de l'interaction bi-manuelle. Un ensemble de tâches élémentaires pertinentes pour l'interaction bi-manuelle et si possible indépendantes du domaine d'application, devra être identifié et permettra de comparer la qualité ergonomique des techniques d'interaction bi-manuelle entre elles au regard d'une tâche.

Enfin nous envisageons d'explorer systématiquement les zones de l'espace 2M qui sont vierges et qui correspondent à aucun système existant. Ceci nous permettra de concevoir de nouvelles formes d'interaction et de les tester expérimentalement.

Plateforme logicielle d'interaction bi-manuelle
Notre système qui repose sur une table augmentée et la manipulation de jetons constitue une véritable plateforme pour développer et tester des interactions bi-manuelles. Tandis que certaines techniques développées sont propres à la manipulation de graphes d'ARN, d'autres comme celles de manipulation des fenêtres et zones de sélection sont génériques. En cela notre système est plus q'une boîte à outils de techniques d'interaction bi-manuelle et constitue une plateforme où l'interaction bi-manuelle s'effectue sur une table avec des jetons ou les doigts. Nous définissons de nombreux travaux liés à cette plateforme outre la programmation à court terme des techniques décrites dans ce mémoire :

- Notre plateforme est dédiée à la manipulation de grands espaces d'information et au travail collaboratif autour de la table. Aussi nous souhaitons montrer son caractère générique en l'appliquant à un autre domaine d'application. Deux domaines sont considérés : la conception de sites web par prototypage et la manipulation de diagrammes UML. Tous deux nécessitent un travail de groupe autour de la table. La conception de sites web nous permettra d'étudier la collaboration et en particulier l'interaction bi-manuelle par deux utilisateurs (une main de chaque utilisateur). La manipulation de diagrammes UML nous fournit un cadre d'étude pour la visualisation de grandes quantités d'information. Ce choix est aussi guidé par le fait qu'il sera facile de trouver des utilisateurs finaux pour les tests.
- Une extension à notre plateforme consiste à ajouter d'autres surfaces d'interaction, par exemple à exploiter les murs pour projeter des informations ou à utiliser un ordinateur de poche comme télécommande pour configurer l'espace informationnel ou comme palette (métaphore du peintre). De nombreux travaux sont à réaliser afin de fournir des techniques d'interaction génériques multi-surfaces.

3 Bilan personnel de l'expérience de recherche de cette année

L'expérience de cette année a été riche d'enseignements et constitue une véritable initiation à la recherche. De plus cette année m'a permis de découvrir le domaine de l'Interaction Homme-Machine.

Interaction Homme-Machine : cycle de vie
Pour la conception de ma plateforme, j'ai appliqué une conception itérative centrée sur l'utilisateur. J'ai pu apprécier la difficulté de l'analyse de l'activité auprès des biologistes, leurs tâches étant principalement exploratoires dans les graphes. J'ai donc décidé d'élaborer des story-boards afin de faciliter les itérations avec les biologistes. Pour cela j'ai dû trouver un dessinateur (étudiant des Beaux Arts de Anger) et travailler en étroite collaboration avec lui pour les scénarios.

Pour le développement, j'ai appliqué des modèles IHM pour la conception logicielle et étudier des mécanismes de programmation systématique de certains composants. J'ai aussi encadré quatre étudiants en RICM2 pendant un mois dont le travail consistait à développer une télécommande sur PDA pour par exemple choisir le graphe projeté sur la table mais aussi changer de mode d'édition du graphe. La facilité d'intégration de leur code dans ma plateforme m'a convaincu de l'apport d'un modèle d'architecture logicielle conceptuelle. D'un point de vue purement techniques de programmation, la complexité résidait dans l'intégration de boîte à

outils comme TclVision (Tcl/Tk) pour le suivi de jetons, Tulip pour la dessin de graphes (OpenGl), au sein d'une architecture d'implémentation répartie avec un serveur.

En conclusion la conception et développement de ma plateforme constituent une véritable expérience de conduite de projet, de l'étape d'analyse de l'activité jusqu'à la réalisation logicielle tout en encadrant une équipe de quatre étudiants. Enfin cette plateforme m'a aussi permis d'appliquer concrètement les modèles, outils et méthodes du domaine de l'Interaction Homme-Machine.

Initiation à la recherche

Outre mes travaux de recherche au sein du laboratoire CLIPS, j'ai eu l'occasion de participer à des réunions de travail. En particulier j'ai présenté mes travaux aux réunions de l'ACI Masse de Données NavGraphe, projet dans lequel s'inscrivent mes travaux mais aussi à une réunion de l'ACI ImpBio projet consacré à l'étude de l'ARN. J'ai aussi effectué des séjours au sein du laboratoire LABRI où travaille D. Auber mon deuxième encadrant. Enfin j'ai rédigé un article long à la conférence UBIMOB 2005 accepté pour publication que j'ai présenté. Cette première expérience de conférences m'a permis de rencontrer de nombreux chercheurs du domaine de l'Interaction Homme-Machine.

Bibliographie

[ALL 83] JAMES F. ALLEN "Maintenaing Knowlege about Temporal Intervals", *communication of the ACM*, 26(11) : p.832-843, 1983

[ATH 84] ATHEMES S. "Adaptabilité et développement de la posture manuelle dans l'écriture: Etude comparative du droitier et du gaucher", *Unpublished memorandum* : Université d'Aix-Marseille II, 1984

[AUB 01] Auber D.. Tulip, a Visualization software for huge graphs. *GD'2001*, 2001.

[BER 99] BERARD F., "Vision par ordinateur pour l'interaction fortement couplée", *Thèse soutenue à Grenoble*, 1999.

[BIE 94] BIER E., STONE M., FISHKIN K., BUXTON W., BAUDEL T.. A Taxonomy of See-Through Tools. *Readings in Human-computer interaction, Morgan-Kaufman,* p. 358-364, 1994.

[BOL 80] BOLT R. "Put That There: Voice and gesture Recognition at the graphics interface". *Computers Graphics, ,* p. 262-270, 1980.

[BOU 01] BOURGEOIS F., GUIARD Y., BEAUDOUIN-LAFON M. "Pan-Zoom coordination in Multi-Scale Pointing". *Proceedings of CHI'2001*, p. 157-158, 2001.

[BUX 83] BUXTON W., "Lexical and pragmatic considerations of input structures", *Computer Graphic*s, 17 (1), p. 31-37, 1983.

[BUX 05] Buxton W., Billinghurst M., Guiard Y, Sellen A, Zhai S., "Human Input to Computer Systems: Theories, Techniques and Technology", *chapitre Two-Handed Input in Human-Computer Interaction*. Unpubhished.

[CAD 94] CADOZ C., "Le geste, canal de communication Homme-Machine. La communication instrumental", *Technique et sciences de l'information (TSI), Hermès ed., vol 13, no.1,* p. 31-61, 1994.

[CAR 83] CARD, MORAN T., NEWEL A. "The psychologie of Human-Computer Interaction", *Lawrence Erlbaum Associates*, 1983.

[CAS 99] CASALTA D., GUIARD Y., BEAUDOUIN-LAFON M."Evaluating two-handed input techniques :Rectangle editiong and navigation", *ACM CHI'99 Conference,* p. 236-237, 1999.

[CRO 94] CROWLEY J., BEDRUNE J., "Integration and Control of Reactive Visual Processes", *European Conference on Computer Vision*, (ECCV-'94), Stockholm, 1994.

[DIE 93] DIETRICH H., MALINOWSKI U..,Kühme T., Schneider-Hufschmidt M. "State of the Art in Adaptive User Interfaces", *[Schneider-Hufschmidt 93]*, pp. 13-48, 93.

[FIT 95] FITZMAURICE G., ISHII H., BUXTON W. "Bricks:Laying the Foundations for Graspable User Interfaces". *Proceedings of CHI'95.* P. 4, 1995.

[FOL 84] FOLEY J., WALLACE V., CHAN P., "The Human Factors of computer Graphics interaction techniques", *IEEE computer Graphics and Applications*, 4(11), p. 13-48, 1984.

[FRO 91] FROHLICH D., "The Design Space of Interfaces, Multimedia Systems", *Interaction and Applications*, *1st Eurographics Workshop*, Stockholm, Suède, Springer Verlag, p. 53-69, 1991.

[GAM 95] GAMMA E., Helm R., Johnson R., Vlissides J, "Design Patterns: Elements of Reusable Object-Oriented Software", ISBN 0-201-63442-2 HardBack 416 pages, 1995.

[GAN 04] GANDIN N., "Techniques d'interaction avec de grands graphes", rapport de DEA, 2004.

[GOU 92] GOURDOL A. , NIGAY L., SALBER D., COUTAZ J. "Two Case Studies of Software Architecture for Multimodal Interactive Systems: VoicePaint and a Voice-enabled Graphical Notebook". *Enginneering for Human-Computer Interaction 92,p. 271-284, 1992.*

[GUI 87] Guiard Y., "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model", *Journal of Motor Behavior*, 19, p. 486-517, 1987.

[HIN 94] Hinckley K., Paush R., Goble J., Kassell N. "Passsive Real-World Interface Props for Neurosurgical Visualization", *CHI'94*, p. 452-458, 1994.

[HUM 87] Hummels C., Smeth G., Overbeeke K., "An intuitive two-handed gestrual interface for computer supported product design", *Proccedings of Gesture workshop'97*, p. 197-208, 1998.

[HUT 86] HUTCHINS E., HOLLAN J. ET NORMAN D., "Direct Manipulation Interfaces", *User Centered System Design, New Perspectives on Computer Interaction*, édité par Norman D., Draper S., Hillsdale, New Jersey : Lawrence Erlbaum Associates, p. 87-124, 1986.

[IHM 91] "Production des participants en ateliers", *IHM'91 Troisième journée sur l'ingénierie des interfaces Homme-Machine*, Dourdan, Hillsdale, p. 11-13, 1991.

[KAB 93]KABBASH, P., MACKENZIE, I.S. & BUXTON, W. "Human performance using computer input devices in the preferred and non-preferred hands". *Proceedings of InterCHI '93, p.* 474-481, 1993.

[KAB 94] KABBASH P., BUXTON W., SELLEN A., "Two-Handed Input in a Compound Task", *In B.Laurel (Ed), The art of Human Computer Interface Design. Addison-Wesley,* p. 307-317, 1990.

[KIR 94] KIRSH D., MAGLIO, P.. "On distinguishing epistemic from pragmatic action". *Cognitive Science 18, 4*, p. 513-549, 1994

[KUR 90] KURTENBACH G., HULTEEN E., "Gesture in Human-Computer Communication", *Proceedings of CHI'94,* p. 417-423, 1990.

[LEG 98] LEGANCHUK, A., ZHAI, S.& BUXTON, W.. "Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study". *Transactions on Human-Computer Interaction,* 5(4), p. 326-359, 1998.

[MAC 90] MACKINLAY J., CARD S., ROBERTSON G., "A Semantic Analysis of the Design Space of Input Devices", *Human Computer Interaction*, Lawrence Erlbaum, Vol. 5, No 2 & 3, p. 145-190, 1990.

[MAR 94] MARTIN, " Etude de la multimodalité fondée sur des types et buts de coopération entre modalités", *Actes des troisièmes journées Internationales sur l'interfaces des Mondes Réels et Virtuels*, 7-11 Février, Montpellier, France. p 97-106, 1994.

[MCL 00] MCLOONE B., KIN POU L., BO-CHIEF Y. "Two Handed Input Using a PDA And a Mouse". *proceedings of CHI'00,* p. 4, 2000.

[MCL 00] MCLOONE H., HINKLEY K., CUTRELL E. "Bimanual Interaction on the Microsoft Office Keyboard". *Interact 2003,* 2000.

[MYE 90] MYERS B., "A New Model for handling Input", *ACM Transactions on Information Systems*, Vol. 8. No. 3, p. 289-320, 1990.

[MYE 00] MYERS B., KIN POU L., BO-CHIEF Y. "Two Handed Input Using a PDA And a Mouse". *Proceedings of CHI'00.* p. 41-48, 2000.

[NIG 93] Nigay L., Coutaz J., "A design space for multimodal interfaces: concurrent processing and data fusion », *Proceedings of CHI'93,* p.172-178, 1993.

[NIG 94] Nigay L., "Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales", *Thèse de l'Université Joseph Fourier*, Grenoble, 1994.

[NIG 96] NIGAY L., Coutaz J.,"Espaces conceptuels pour l'interaction multimédia et multimodale", *Journal TSI, spécial Multimédia et Collecticiel*, AFCET &Hermès Publ. 15(9), p. 96, 1996

[PFA 92] Pfaff G.,"The uims workshop tool developers : A metamodel for the runtime architecture of an interactive systems", sigchi bulletin, 24, 1, p.32-37, 1992.

[PFA 92] Pfaff G.,"User interface management systems".

[REN 04] Rennevier P.,"Systèmes mixtes collaboratifs sur supports mobiles : conception et réalisation", *thèse à l'unversité Joseph-Fourier*, Grenoble, 2004

[SCO 01] SCOTT MACKENZIE I., GUIARD Y., "The Two-Handed Desktop Interface: Are we Yet?", Published in Extended abstracts of CHI'01, p. 351-352, 2001.

[SCH 93] SCHNEIDER-HUFSCHMIDT M., Kühme T., Mallinowski U., "Adaptative User Interfaces, Principles and Practice", editeurs, Bullinger H-J, dans la série Human Factors in Information Technologie 10 paru chez Elsevier Science Plublishers B.V, Horth-Holland, 1993.

[VER 01] VERNIER F., "La multimodalité en sortie et son application à la visualisation de grandes quantités d'information", *Thèse de l'Université Joseph Fourier*, Grenoble, 2001.

[YEE 04] YEE K..Two-handed interaction on a Tablet Display. Conference on Human Factors in computing Systems. Vienna, Australia, p. 1493-1496, 2004.

[ZHA 97] ZHAI S., SMITH S., SELKER T.. Improving Browsing Performance: A study of Four Input Devices for Scrolling and Pointing. *Proceedings of CHI'9.* P. 286-292, 1997.

[KRU 91] KRUEGER M., *Artificial Reality II*, Addison Wesley, 1991.

[WEL 93] WELLNER P., "Interacting with paper on the Digital Desk", *Communications of the ACM*, Vol. 36, No. 7, p. 87-97, 1993.

Tables des figures

FIGURE 1 : DEUX VUES DU MEME TEXTE ECRIT A LA MAIN [GUIARD 87]. LA PAGE DU COTE GAUCHE DE LA FIGURE 1 MONTRE CE QUE LE SCRIBE A ECRIT SUR LE PAPIER. LA PAGE DU COTE DROIT FIGURE 1MONTRE CE QUI EST APPARU SUR LA COPIE AU CARBONE PLACEE SUR LA TABLE EN DESSOUS DE LA FEUILLE ECRITE AVEC LA MAIN DOMINANTE. L'ANGLE DE L'ECRITURE, COMME LE RECOUVREMENT DES LIGNES EST UNE CONSEQUENCE DU CHANGEMENT DE CONTEXTE EFFECTUE PAR LA MAIN NON DOMINANTE. 18
FIGURE 2 : DISPOSITIF PHYSIQUE VU COMME TRANSDUCTEUR DE TROIS PROPRIETES PHYSIQUES [BUXTON 83]. ... 25
FIGURE 3 : CLASSIFICATION DE QUELQUES DISPOSITIFS DANS LA TAXONOMIE DE BUXTON 26
FIGURE 4 : BOUTON CRANTE 27
FIGURE 5 : LA TAXONOMIE DE FOLEY DIRIGEE PAR LES TACHES GRAPHIQUES [FOLEY 84]. 29
FIGURE 6 : LES QUATRE NIVEAUX D'ABSTRACTION (DISPOSITIF, LANGAGE, TACHE, BUT) DEFINI DANS [NIGAY 94] ET ETENDU AVEC [DIETRICH 93]. 32
FIGURE 7 : ESPACE DE CONCEPTION DES INTERFACES EN ENTREE DE FROLICH. 33
FIGURE 8 : L'ESPACE DE CONCEPTION MSM ([NIGAY 96]).. 34
FIGURE 9 : LES DIFFERENTS CAS D'USAGE EN ENTREE DE MODALITES (SEQUENTIELLE/PARALLELE ;FUSION/FISSION) [NIGAY 96]... 36
FIGURE 10 : CARE ETENDUES. APPLICATION DES SCHEMAS DE COMPOSITION AUX CINQ ASPECTS DE COMPOSITION [VERNIER 01]. 38
FIGURE 11 : REALISER LES SOUS-TACHES A ET B EN UTILISANT UNE APPROCHE A UNE MAIN [BUXTON 05].......... 42
FIGURE 12 : REALISER LES SOUS-TACHES A ET B EN UTILISANT UNE APPROCHE A DEUX MAINS [BUXTON 04]...... 43
FIGURE 13 : REALISER LES SOUS-TACHES A ET B EN INTRODUISANT DU PARALLELISME [BUXTON 04]................. 43
FIGURE 14 : REALISER LES SOUS-TACHES A ET B AVEC UN COUT COGNITIF ELEVE [BUXTON 04].......................... 43
FIGURE 15 : REPRESENTATION DES CINQ NIVEAUX D'ABSTRACTION POUR L'ETUDE DE LA MULTIMODALITE TIREE DU SCHEMA DE [NIGAY 94] COMPLETE PAR [DIESTRICH 93] ET [CADOZ 94].. 45
FIGURE 16 : CARACTERISATION DES GESTES DE CHAQUE MAIN : DES RELATIONS ENTRETENUES ENTRE CES CARACTERISTIQUES EMERGE UNE INTERACTION BI-MANUELLE OU NON.. 46
FIGURE 17 : LES SEPT RELATIONS D'ALLEN [ALLEN 83] 47
FIGURE 18 : SEQUENCE DE SOUS-TACHES AVEC UN TEMPS DE TRANSITION (PARTIE GAUCHE) ET SANS TEMPS DE TRANSITION (PARTIE DROITE). EXEMPLE : SOUS-TACHE B = CHOIX DE LA COULEUR DANS UNE PALETTE ET SOUS-TACHE A = COLORIAGE DE LA COULEUR COURANTE. 48
FIGURE 19 : CONCOMITANCE (CHEVAUCHEMENT) DES SOUS-TACHES A ET B. PAR EXEMPLE LES SOUS-TACHE B = DEFILEMENT ET SOUS-TACHE A = SELECTIONNER POUR UN UTILISATEUR EXPERT. 48
FIGURE 20 : COÏNCIDENCE (INCLUSION) DES TACHES A ET B DANS L'INTERACTION A DEUX MAINS. PAR EXEMPLE SOUS-TACHE B = ENFONCEMENT D'UNE TOUCHE SPECIALE AU CLAVIER ET SOUS-TACHE A = SELECTION MULTIPLE AVEC LA SOURIS.. 49
FIGURE 21 : PARALLELISME DES SOUS-TACHES A ET B DANS L'INTERACTION A DEUX MAINS. PAR EXEMPLE POUR FAIRE AVANCER EN LIGNE DROITE UN CHAR DANS UN JEU DE REALITE VIRTUELLE, SOUS-TACHE B = APPUI DE L'ACCELERATEUR POUR LE CONTROLE DE LA VITESSE D'UNE CHENILLE D'UN CHAR ET SOUS-TACHE A = APPUI DE L'ACCELRATEUR POUR LE CONTROLE DE LA VITESSE DE L'AUTRE CHENILLE D'UN CHAR............. 49
FIGURE 22: ESPACE DE CONCEPTION 2M POUR L'INTERACTION BI-MANUELLE. 51
FIGURE 23 : ILLUSTRATION DU CAS DE DEPENDANCE FUSIONNELLE : DEPENDANCE D'UN CHAR DANS UN JEU. A CHAQUE MAIN EST ASSOCIE L'ACCELERATEUR D'UNE CHENILLE.. 52
FIGURE 24 : ILLUSSTRATION DU CAS DE DEPENDANCE TEMPORELLE ET FUSIONNELLE : EDITEUR DE DESSIN BI-MANUEL. L'UTILISATEUR FAIT GLISSER LA ZONE D'EDITION AVEC SA MAIN NON DOMINANTE ET DESSINE UN TRAIT AVEC SA MAIN DOMINANTE. 53
FIGURE 25 : SYSTEME BI-MANUEL AVEC UN JOYSTICK DANS LA MAIN NON DOMINANTE ET UNE SOURIS DANS LA MAIN DOMINANTE PUR LA TACHE DE SCROLLER ET POINTER [ZHAI 97]. 55
FIGURE 26: SYSTEME BI-MANUEL AVEC UN PDA DANS LA MAIN NON DOMINANTE ET UNE SOURIS DANS LA MAIN DOMINANTE POUR LA TACHE DE SCROLLER ET POINTER [MYERS 00] 55
FIGURE 27 : CLAVIER MICROSOFT AUGMENTE AVEC UNE ROUE POUR SCROLLER [MCLONNE 03]....................... 56
FIGURE 28 : TACHE DE DESSIN [YEE 04]. IL EST POSSIBLE DE SELECTIONNER LA COULEUR OU DEPLACER LA FEUILLE VIRTUELLE AVEC LA MAIN NON DOMINANTE ET DESSINER AVEC LA MAIN DOMINANTE. 57
FIGURE 29 : TOOLGLASS [BIER 04].. 58
FIGURE 30 : CLAVIER XEROX AVEC DES RACCOURCIS SUR LA GAUCHE POUR LA MAIN NON DOMINANTE.............. 58
FIGURE 31 : UNE INTERFACE PALPABLE AVEC DES BRIQUES SAISISSABLES [FITZMAURICE 95]. 59
FIGURE 32 : EXEMPLE DE MANIPULATION DES BRIQUES (TRANSLATION, ROTATION) DANS UNE INTERFACE PALPABLE [FITZMAURICE 95]... 59
FIGURE 33 : UNE INTERFACE PALPABLE DE VISUALISATION POUR LA NEUROCHIRURGIE [HINCKLEY 94]. 60

FIGURE 34 : LA NOTATION HTA 70
FIGURE 35 : L'APPLICATION NAVGRAPHE 70
FIGURE 36 : UTILISER L'APPLICATION 71
FIGURE 37 : TRAVAIL SUR LE GRAPHE 71
FIGURE 38 : TACHE MANIPULER LE GRAPHE 72
FIGURE 39 : TACHE EDITER LE GRAPHE 72
FIGURE 40 : TRAVAIL SUR LE SOUS –GRAPHE.. 73
FIGURE 41 : DEPLACEMENT A L'AIDE D'UN JETON D'UN FENETRE WINDOW2M.. 74
FIGURE 42 : ROTATION ET ZOOM D'UNE FENETRE WINDOW2M A L'AIDE D'UN JETON. 74
FIGURE 43 : ROTATION ET ZOOM DE LA FENETRE A L'AIDE DES DEUX MAINS. 75
FIGURE 44 : DEPLACEMENT ET REDIMENSIONNEMENT DE LA FENETRE A L'AIDE DES DEUX MAINS. 76
FIGURE 45 : DEFORMATION DU GRAPHE GRACE AU MODELE DE FORCE 77
FIGURE 46 : RECONSTRUCTION D'HELICE D'ARN.. 77
FIGURE 47 : TABLE AUGMENTEE [BERARD 99] APPLIQUEE AU DOMAINE DE LA BIOLOGIE. 79
FIGURE 48 : ARCHITECTURE PAC-AMODEUS 81
FIGURE 49 : ARCHITECTURE DE L'APPLICATION QUI SUIT LE MODELE PAC-AMODEUS 82
FIGURE 50 : HIERARCHIE D'AGENTS PAC DU CONTROLEUR DE DIALOGUE. 83
FIGURE 51 : MODELE UML D'UN AGENT PAC... 86
FIGURE 52 : DEMARCHE DE TRAVAIL ET CHAPITRES DU MEMOIRE.. 90

Chapitre VIII Annexes

La première annexe présente l'outil de capture de jetons sur la table augmentée TCLVision et l'application que nous avons développées qui récupère ces événements en C++.

La seconde annexe explicite tout ce qui est nécessaire pour installer et développer l'application.

La troisième annexe présente le story-board élaborée en collaboration avec romain Boussard.

La dernière annexe présente quelques points de l'application développée par les étudiants que nous avons encadrés.

1 *CVision/TCLVision*

1. 1 TCLVision

TCLVision est un outil de vision par ordinateur élaboré par François Berard (francois.berard@imag.fr). Il permet à l'aide d'une caméra de capter la position de jetons sur une table. Il indique également si un jeton apparaît, disparaît ou se déplace. Ce logiciel est présent sur l'ordinateur POUCET.

Nous détaillons ici la gestion des événements avec TCLVision. La position des jetons est reconnue par le système de vision (TCLVision) : Ces évenements sont traduits en événements utilisateur qui arrive sur une socket TCP :

- à l'apparition d'un jeton sur la table.
- quand il y a déplacement d'un jeton .
- à la disparition d'un jeton.

Tous les messages émis et recus par TCLVision sont codés en NetworkByteOrder (BigEndian).

Les messages émis par TCLVision sont les suvants :

- ACK (10) : Un acquittement à chaque appel à TCLVision
- EVT (3) : l'envoi des différents paramètres d'un jeton. Ce dernier est composé :
  - D'un code évenement (MOTION (0), APPEAR (1), DISAPPEAR (2))
  - L'identifiant du jeton (un entier)
  - La partie entière de la coordonnée x du jeton (un entier)
  - La partie décimale de la coordonnée x multiplié par 1000 du jeton (un entier)
  - La partie entière de la coordonnée y du jeton (un entier)
  - La partie décimale de la coordonnée y multiplié par 1000 du jeton (un entier)
  - L'étiquette temps, en millisecondes de l'évenements

Les messages qui peuvent être envoyés par la socket à TCLVision sont les suivants :

- START (1) : pour lancer TCLVision (le serveur émet un ACK si la commande a été comprise).

- STOP (2) : pour éteindre TCLVision (le serveur émet un ACK si la commande a été comprise).
- STARTEVENT (5) : commencer l'écoute d'événements (le serveur émet un ACK si la commande a été comprise).
- STOPEVENT (6) : arrêter l'écoute d'événements (le serveur émet un ACK si la commande a été comprise).
- CALIBRAGE. (7): permet d'ajuster les paramètres de la caméra. (le serveur émet un ACK si la commande a été comprise).
    - Nombre de points que l'on souhaite émettre pour le calibrage généralement 9.
    - La liste des coordonnées des points (coordonnées exprimées par des entiers) au format d'une chaine de caractère« x1 y1 x2 y2 …x9 y9 ». (le serveur émet deux ACK si la commande a été comprise).

TCLVision émet des acquittements à la réception des messages.

1. 2 CVision

Nous proposons d'élaborer la partie client de TCLVision en C++ qui soit réutilisable. Pour cela nous avons écrit deux classes :

- La classe Token qui possède tous les attributs nécessaires (codeEvenement, id, x, y, temps) et les ascesseurs correspondants.
- la classe CVision capable de se connecter au programme TCLVision par socket, de le lancer et d'écouter les différents événements. La partie réseau est totalement transparente pour l'utilisateur et repose sur la librairie de Lionel Balme (lionel.balme@imag.fr) :SlimLib.

Les méthodes de CVision sont donc les suivantes :

Start() : pour lancer le serveur TCLVision.
Stop() : pour arrêter le serveur TCLVision.
StartListen() :pour commencer à écouter les événement TCLVision
StopListen() :pour arrêter à écouter les événements TCLVision
Calibrage() : pour calibrer TCLVision

En fonction des événements lus sur la socket, le programme met à jour la structure de données contenant toutes les informations des jetons présents sur la table (stocké dans une "map" pour permettre l'indexation directe par l'identifiant du Jeton).

CVision est incorporé dans un thread afin de ne pas bloqué le programme parent et pour des soucis de performance.

2 Installation et développement

Nous détaillons ici les installations nécessaires pour faire tourner l'application.

2. 1 Installer Linux

Dans le cadre de notre étude, nous avons utilisé Fedora Core 3.

2. 2 Installer la carte graphique nvidia

L'installation de la carte graphique n'est pas toujours évidente. Il est conseillé de bien connaître linux pour le faire. Les étapes sont les suivantes :

- Télécharger sur le site de nvidia le driver pour linux. Il en existe qu'un pour toutes les cartes gforce.
- Ouvrir une console à l'aide de la commande ctrl alt f2.
- Se connecter en root.
- Arrêter le serveur X responsable de l'affichage avec la commande :
  > Init 3
- Lancer le pilote nvidia avec la commande
  > Bash <nomDuDriverNvidia.bin> Répondre oui aux différentes questions.
- Une fois le driver installé, il faut modifier le fichier xorg.conf dans /etc/X11/. Editer le fichier en mode console avec emacs par exemple.
- Commenter la ligne : Load « dri ». Remplacer Driver « nv » par Driver « nvidia ». Voici le résultat que vous devez obtenir.

```
Section "Module"
      o  Load  "dbe"
      o  Load  "extmod"
      o  Load  "fbdevhw"
      o  Load  "glx"
      o  Load  "record"
      o  Load  "freetype"
      o  Load  "type1"
#    Load  "dri"
EndSection

…

Section "Device"
      o  Identifier  "Videocard0"
      o  Driver      "nvidia"
      o  VendorName  "Videocard vendor"
      o  BoardName   "NVIDIA GeForce 4 MX
         (generic)"
EndSection
```

- Redémarrer Linux pour que la carte graphique soit prise en compte.

Si la carte graphique n'est pas correctement installée, notre programme ne fonctionnera pas correctement et sera très lent (temps de latence trop élevé pour l'interaction). De plus, nous utilisons une commande openGl drawBuffer(AUX1) qui ne fonctionnera plus sans l'installation de la carte. Donc au lieu d'afficher dans ce buffer, openGL l'affichera à l'écran.

2. 3 Installer Tulip

Les sources de tulip sont récupérables à http://sourceforge.net/projects/auber/
D'autres informations utiles sont présentes sur http://www.tulip-software.org/
Pour installer Tulip, les étapes sont les suivantes :

- Récupérer l'archive tulip-2.0.2.tar.bz2 qui contient les sources de tulip.
- Décompressez l'archive grâce aux outils bunzip2 et tar –xvf.
- Dans le répertoire tulip-2.0.2 qui a été créé, tapez « ./configure » : cette commande permet de vérifier que Tulip trouve tout ce qu'il a besoin sur votre ordinateur.
- Saisir la commande « make » : cette commande permet de compiler Tulip. Il faut environ 20 minutes pour compiler.
- Une fois la compilation terminée, il s'agit de créer les librairies qu'il faudra insérer dans le programme « tulip », « tulip-ogl », « tulip-qt ». Pour cela tapez « make install ». Les répertoires « /usr/loca/bin », « /usr/local/include », « /usr/local/lib » et « /usr/local/share/ » seront alors modifiés.

- L'installation de Tulip est alors terminée. Exécuter la commande « Tulip ».

Pour connaître les informations sur Tulip vous pouvez utiliser la commande suivante :
tulip -config --libs pour savoir quelles sont les librairies Tulip et où elles sont installés.
tulip -config --version pour connaître la version
tulip -config –pluginpath pour savoir où sont les plugins.

Afin de pouvoir utiliser les différentes librairies créées, vous devez impérativement indiquer au système où elles se trouvent : Pour cela éditer le fichier .bash_profile qui se trouve à la racine de votre compte et ajouter :

LD_LIBRARY_PATH= /usr/local/lib/ :$LD_LIBRARY_PATH
Export LD_LIBRARY_PATH

La variable d'environnement LD_LIBRARY_PATH contient le chemin pour les différentes librairies, nous avons donc ici rajouter /usr/local/lib à ce path.

2. 4 Installer Eclipse

Nous avons travaillé sous Eclipse avec le module CDT qui permet de faire du C ou du C++. Nous avons choisi cet environnement de développement car il est capable d'importer du source C++ et de générer automatiquement un makefile. Eclipse est multi plateforme. Il est donc plus facile avec cet environnement de faire passer son code sous windows si besoin était.

Pour installer Eclipse, vous devez d'abord installer la dernière version de la machine virtuelle java que l'on trouve sur le site de Sun. Par défaut sous linux, vous pouvez exécuter du java grâce à gcc. Il faut maintenant lui indiquer que vous voulez utiliser la commande java de la machine virtuelle que vous venez d'installer. Pour cela, vous devez éditer le fichier .bash_profile à la racine de votre compte, et modifier la variable d'environnement PATH.

PATH=/bin/java/jsdk1.5/bin/ :$PATH

ATTENTION, l'ordre ici est important. Si vous inversez, le java de gcc sera le premier trouvé et donc le premier exécuté.

Vous pouvez maintenant installer Eclipse. Nous avons utilisé la version 3.0.

2. 5 Installer CDT (module C et C++ pour eclipse)

CDT est un plugin Eclipse qui permet de gérer des programmes en C et en C++. Pour installer le CDT, lancer Eclipse. Dans la barre de menu, cliquer sur :

`Help > Software Updates > Find and Install ...`

Cocher *Search for new features to install* puis cliquer sur *Next*. Cliquer sur *New Remote Site*.... Il faut alors saisir un nom, par exemple `Eclipse CDT` et saisir l'url `http://update.eclipse.org/tools/cdt/releases/new` (pour *CDT 2.x*)

Ou

`http://update.eclipse.org/tools/cdt/updates/release` (pour *CDT 1.x*).

Attention :

CDT 2.0.0, *2.0.1*, et *2.0.2* s'utilisent avec *Eclipse 3.x* et *CDT 1.1*, *1.2*, et *1.2.1* s'utilisent avec *Eclipse 2.x* **!**

Ensuite valider en cochant la version du CDT désiré (ex: *CDT 2.0.2*).

2. 6 Installer glut, glu et opengl

Les librairies associées sous linux sont :

GL, GLU et glut. Vous devrez donc les rajouter aux propriétés de votre projet eclipse.

Il y a un problème avec la librairie glut que l'on peut contourner en allant dans le repertoire :

/lib. Vous devez ensuite renommer le fichier libglut.so.2 en libglut.so afin qu'il puisse être reconnu comme une librairie. Vous devez également télécharger le fichier glut.h car ce fichier est demandé pour pouvoir compiler.

2. 7 Installer SlimLib

Slimlib est une librairie réseaux. Son intérêt est sa portabilité (linux, mac, windows). Vous devez demander à Lionel Balme (lionel.balme@imag.fr) le code pour Slimlib. Votre code doit inclure les fichiers .h pour la compilation et vous devez fournir une librairie.

2. 8 Installer le projet navGraphe

Sous Eclipse, importer le répertoire navGraphe que vous aurez installé à la racine de votre compte. Les répertoires à inclure sont :

~/navGrapheR/include

~/navGrapheR/PRESENTATION/BiToolKit/include

Les librairies à inclure sont :

- Glut, GLU, openGL

- Tulip, tulip-ogl, tulip-qt (facultatif)
- slimLib

Les paths pour les librairies sont :

- /usr/local/lib

3 Story-Board

Les story-Boards élaborés en collaboration avec le dessinateur Romain Boussard.

4 Projet RICM2

Vous trouverez le sujet du projet, ainsi que quelques documents qui nous ont été remis. La documentation du projet est disponible à l'adresse : http://iihm.imag.fr/bailly/pda/

Résumé

L'homme utilise naturellement au quotidien les deux mains pour effectuer des tâches dans le monde physique. Dans le contexte de l'informatique pervasive qui vise la fusion des mondes physique et numérique, l'interaction bi-manuelle semble donc une solution à explorer. Dans mon étude, je propose un nouvel espace de conception pour l'interaction bi-manuelle, noté 2M. Cet espace organise dans un canevas unificateur des résultats issus d'études sur l'interaction multimodale et bi-manuelle. D'une part, nous enrichissons les travaux sur l'interaction multimodale pour prendre en compte le rôle des deux mains ; d'autre part nous étudions l'interaction à deux mains au regard de la multimodalité (une modalité par main) pour prendre en compte les aspects temporels et de dépendance de la multimodalité lors de la conception de techniques d'interaction bi-manuelle.

Le pouvoir descriptif de l'espace 2M est évalué en étudiant plusieurs systèmes d'interaction bi-manuelle existants. J'étudie aussi la capacité de l'espace 2M à guider la conception de formes d'interaction bi-manuelle dans le contexte du projet NavGraphe dédié à la manipulation de grandes structures biologiques comme l'ARN (acide ribonucléique), après avoir étudié les besoins des utilisateurs finaux, les biologistes. Les résultats de conception ont enfin été développés pour définir un système interactif bi-manuel multi-surface pour explorer et manipuler des grands graphes.

Mots Clés
Interaction Bi-Manuelle, Multimodalité, Espace de Conception.